Basicskolan — lektion nr 1

Den här veckan startar jag med en åtta veckors kurs i grundläggande Basicprogrammering. Jag kommer också varje vecka försöka att ha ett program i matematik, som många har efterlyst.



Hem DATORER

Av Christian Lindeberg

Naturligtvis kommer jag också att svara på brevfrågor. Jag börjar med en brevfråga och en läsare från S Sandby har en del frågor, som är gemensamma för många av er. Läsaren har en VIC-64 med bandspelare och han har följande frågor. Han hittar inte tecknet för £ på sin dator. Anledningen till att du inte hittar tecknet beror på att man behövde den platsen när datorn skulle utrustas med å, ä och ö. Ö finns i stället för £. Du skriver tyvärr inte varför du vill ha fram ett pundtecken så jag kan inte ge dig en alternativlösning.

Registerprogram

Nästa problem är ett personregisterprogram skrivet för disk. I programmet finns en mängd rader med OPEN-satser, som läsaren försökt att översätta till bandspelare utan att lyckas. Jag förutsätter att programmet är med sekventiella filer. Raden måste se ut så här t ex

1000 OPEN 2,8,2,"NAMN,S,R" eller
2000 OPEN 2,8,2,"§:NAMN,S,W"

Efter OPEN finns tre siffror och de kan vara annorlunda i ditt program. Huvudsaken är att det står en 8 i mitten. Detta är nämligen datorns siffra för diskenheten. Efter NAMN, står bokstaven S, som betecknar sekventiell fil och nästa bokstav talar om ifall datorn läser eller skriver till disken. R betyder READ = läsa och W betyder WRITE = skriva. Detta system är annorlunda till bandspelare, där man använder en 1 eller 0 för att läsa eller skriva. Raderna till bandspelare blir i stället så här:

1000 OPEN 1,1,0,"NAMN" för att läsa och
2000 OPEN 1,1,1,"NAMN" för att skriva till band.

Du måste tänka på att bandspelaren måste vara framspolad till rätt ställe.

Utan ljuspenna

Läsaren vill också ha ett program, där han kan rita med datorn utan att använda ljuspenna. Det är svårt att göra ett enkelt program för att rita. Det finns en del utmärkta program att köpa, t ex "Blazing Paddles", "Paint Magic", "Koala Painter" och "Doodle". Till alla dessa program kan du använda joystick. Min favorit av dessa program är "Doodle".

Sista frågan gäller namnet på programmet, som jag talade om i artikeln den 4 januari. Det var "Little Computer People", som finns både till disk och bandspelare. Diskversionen har många fler möjligheter. Det är bara i diskversionen, som man kan spela spel med datagubben.

Nu över till Basicskolan lektion ett.

Den viktigaste knappen på datorn heter RETURN, ENTER eller NEWLINE beroende på vilken dator du använder. Det är denna knapp, som ger signal till datorn att utföra sina order. Det vanligaste felet hos nybörjare är att man har glömt att trycka på RETURN-knappen.

Två instruktioner

Den här gången skall vi använda två stycken instruktioner nämligen PRINT och GOTO. Print använder man när datorn skall skriva ut något på skärmen. Testa genom att skriva PRINT "ARBETET" och tryck därefter på RETURN. Nu skriver datorn ut ordet ARBETET. Efter PRINT ser du att citationstecken finns runt ordet ARBETET. Citationstecken måste alltid vara med då man vill ha utskrivet bokstäver och tecken. Finns inte något citationstecken så tror datorn att det är siffror den skall skriva ut. Pröva genom att skriva PRINT ARBETET och tryck på RETURN. Datorn svarar nu med siffran noll. Ytterligare förklaring kommer i nästa artikel.

Radnummer

För att kunna köra ett program i Basic så måste datorn veta i vilken ordning den skall utföra saker och ting. Detta hjälper man datorn med genom att använda radnummer mellan 0 och 65535. Normalt brukar man använda sig av var tionde siffra med början på siffran 10. Alltså 10, 20, 30 och så vidare. Nästa instruktion vi skall använda är GOTO, som är ett ovillkorligt hopp. Datorn hoppar till den rad, som står efter instruktionen GOTO.

Skriv följande lilla program.

```
10 PRINT "ARBETET"
20 GOTO 10
```

Glöm inte att efter varje rad trycka på RETURN annars lagras inte raden i datorns minne. När du är klar skriver du kommandot RUN som betyder att datorn skall utföra ditt program. Om du har gjort rätt så får du längst till vänster en rad där datorn skriver ut ordet ARBETET.

Ändra tecken

Jag hoppas att du har läst så mycket i din instruktionsbok att du vet hur man stoppar ett program. Om du har en VIC så tryck på RUN/STOP, annars tryck på den tangenten, som heter BREAK. Skulle du ha en dator, som arbetar med något operativsystem så tryck på CTRL och C. Skriv om rad 10 på dessa två olika sätt och se den stora skillnad du får genom att bara ändra ett tecken.

10 PRINT "ARBETET", och glöm inte RETURN. Du behöver inte skriva om rad 20. När du testat detta så skriv.

10 PRINT "ARBETET"

Basicskolan fortsätter nästa vecka.

Till slut ett litet program om fakultet, alltså tal, som multipliceras i en följd. Fakultet skrives med tecknet !, så 3! = och 41! = 1*2*3*4 o s v. Se fter hur snabbt fakulteten ökar från 1! och uppåt. Försök lista ut programmet själv annars skriv till mig.

```
10 PRINT CHR$(26)
20 T=1
30 PRINT"VILKEN FAKULTET
   VILL DU HA";
40 INPUT N
50 FOR I=1 TO N
60 PRINT I;
70 IF I<N THEN PRINT "*";
80 T=T*I
90 NEXT I
100 PRINT"=";T
```

GET — en instruktion som vållar problem

Det är svårt med dataprogram. I förra veckans artikel lyckades alla gångertecken bli x-tecken. Ni, som inte lyckades med programmen så leta efter x-en och ändra till gångertecknet, som är stjärnan eller asterisken på din dator.



Av Christian Lindeberg

I veckan så var jag på datamässan i Malmö, egentligen så var den för kontorsdatorer, men jag visste att man skulle visa den nya AMIGAN från Commodore. Under den tid då jag var på mässan så var det folk på två ställen det ena var kring en trollkonstnär? Och den andra folksamlingen var kring AMIGAN. I reklambladet om AMIGAN så skriver man "Den makalösa Amigan".

Avancerad

Jag måste hålla med att detta var en av de mest avancerade datorer jag har sett. Det är ingen dator för hemmabruk, men en fantastisk dator med många möjligheter. Det är en 16-bitars dator, vilket gör den snabb plus att den har tre stycken hjälp-processorer, som hjälper till att flytta data mellan olika RAM-positioner. Utan att behöva ta hjälp av huvudprocessorn.

Tre program

Man kan köra tre program samtidigt, vilket är en stor fördel speciellt för dem som arbetar med ordbehandling, kalkylering och registerprogram. Att man kan göra detta beror på att Amigan har en mycket avancerad fönsterhantering. På skärmen kan man alltså se alla tre programmen på samma gång. Det som jag var mest imponerad av är grafiken, som kunde hantera 4096 färger i högupplösning. Även ljudmöjligheterna var mycket imponerande med fyra kanaler och separata stereoutgångar.

Varje kanal kan hantera 8 till 9 oktaver, vilket är mer än många stereoanläggningar kan klara av. Det skulle vara positivt om Amigan kom ut i en lite billigare modell till hemdatorsidan, där det finns ett stort intresse för både grafik och ljud. Komplett kostar Amigan ca 20.000:-.

Jag har fått flera frågor om samma sak, nämligen användning av instruktionen GET. GET är mycket användbar när man skall välja olika alternativ. Jag skall visa två programexempel med och utan GET.

GET och INPUT

Först en del om GET jämfört med INPUT. GET känner bara av ett tecken, som måste hanteras som en sträng, medan INPUT kan hantera flera tecken både numeriska och sträng-variabler. GET känner av tangentbordet och reagerar på nedtryckt tangent. Det finns ytterligare en variation man kan använda i valsituationer nämligen ON GOTO och ON GOSUB.

Skillnaden är att i första fallet hoppar datorn till ett radnummer och fortsätter där, men i det andra så hoppar datorn till en subrutin och återvänder till radan efter där den hoppade ut. I mina två programförslag har jag i det första använt mig av den vanliga med INPUT. I den andra har jag använt ON GOTO på två olika sätt.

I INPUT-exemplet är det inte så mycket att kommentera, men i exemplet med ON GOTO tror jag att rad 230 måste förklaras. Ni som följt med i Basic-skolan kommer kanske ihåg avsnittet om IF THEN-satser. Om du har glömt så testa följande på din dator skriv följande:

PRINT 5=5

Datorn svarar nu -1 och skriver du så här:

PRINT 4=5

Datorn svarar med 0. Vad man testar är om något är sant eller inte. Datorn ger -1 när något är sant och 0 när något är falskt. Om du ser i rad 230 så finner du två jämförelser antingen om strängen A är lika med J eller om strängen A är lika med N. Om nu ett J trycks ner så blir första uttrycket -1 som multipliceras med 2 vilket blir -2 till detta adderas nästa uttryck som blir Ø eftersom J var nedtryckt och till detta lägger vi 3 och svaret blir 1.

Datorn hoppar till första radnumret i ON GOTO-satsen, vilket är 500. Om du i stället trycker ner N, så blir uträkningen följande. Första villkoret blir 0 och multiplicerat med 2 blir det fortfarande 0. Nästa uttryck blir sant och lika med -1 och till detta adderar vi 3 och får svaret 2. Nu hoppar datorn till rad 10.

Mindre plats

Om du trycker ner en annan tangent så blir båda uttrycken falska och 0, vilket gör att svaret blir 3 och datorn ligger kvar i rad 230. Samma sak inträffar om ingen tangent är nedtryckt. En fördel med skrivsättet med GET-satsen är att programmen tar mindre plats.

Ett par kommentarer till GET är att en del datorer tycker inte om att man skriver på detta sätt och ger ett felmeddelande "type mismatch", vilket betyder att datorn tycker att man har blandat ihop numeriska variabler med strängvariabler. ABC-datorer har ett GET-kommando, som inte fungerar på det vanliga sättet så där kan man inte alls använda rad 230.

Skriv och berätta om dina egna tricks för att göra programmen kortare och snabbare.

Olika språk

Det har kommit många frågor om olika språk och hur man använder dem. I många skolor har man till exempel börjat att använda Comal, som är ett språk kombinerat mellan Basic och Pascal. Jag tänkte återkomma i en hel artikel och diskutera olika språk till våra hemdatorer. Det finns mycket att få till våra hemdatorer.

```
1Ø PRINT CHR¤(26)
2Ø PRINT
3Ø PRINT"MENY"
4Ø PRINT"VAL 1."
5Ø PRINT"VAL 2."
6Ø PRINT
7Ø INPUT A
8Ø IF A=1 THEN 2ØØ
9Ø IF A=2 THEN 3ØØ
1ØØ GOTO 1Ø
2ØØ PRINT CHR¤(26)
21Ø PRINT"VAL 1"
22Ø PRINT"VILL DU SLUTA J/N"
23Ø INPUT A¤
24Ø IF A¤="J" THEN END
25Ø IF A¤="N" THEN 1Ø
26Ø GOTO 22Ø
3ØØ PRINT CHR¤(26)
31Ø PRINT"VAL 2"
32Ø END
```

```
1Ø PRINT CHR¤(26)
2Ø PRINT
3Ø PRINT"MENY"
4Ø PRINT"VAL 1."
5Ø PRINT"VAL 2."
6Ø PRINT
7Ø INPUT B
8Ø ON B GOTO 2ØØ,3ØØ,1Ø
2ØØ PRINT CHR¤(26)
21Ø PRINT"VAL 1"
22Ø PRINT"VILL DU SLUTA J/N"
23Ø GET A¤:ON ((A¤="J")*2+(A¤="N")+3)
GOTO 5ØØ,1Ø,23Ø
3ØØ PRINT CHR¤(26)
31Ø PRINT"VAL 2"
32Ø END
5ØØ END
```

Gör ditt eget kryptoprogram



Av Christian Lindeberg

Denna vecka kommer att handla om logiska operatorer. Det har kommit många frågor om hur de fungerar. De logiska operatorerna är bland annat AND, OR och NOT. På en del basic-dialekter finns också XOR. Jag skall visa hur man skriver om en XOR med hjälp av AND, OR och NOT.

Det har också kommit en fråga från en läsare i Trelleborg om kryptoprogram. Hans fråga passar bra in i dagens artikel eftersom krypto görs mycket klurigt med hjälp av XOR.

Först till de vanliga operatorerna och jag börjar med AND. AND kan som de andra användas på två sätt. Det vanligaste är att man använder AND för att två villkor skall uppfyllas. Till exempel IF A=6 AND B=7 THEN 200.

Datorn testar

Datorn testar om både A och Bär uppfyllda och i så fall hoppar datorn till rad 200.

Det andra sättet är att testa om en viss bit i en byte innehåller ett visst värde exempelvis om man vill testa om adressen 32500 innehåller en etta i 6:e biten den som har vikten 64. Vikterna är från höger 1, 2, 4, 8, 16, 32, 64, 128 och numeras 0, 1, 2, 3, 4, 5, 6 och 7. I basic skriver man då så här:

IF (PEEK (32500) AND 64) = 64 THEN 300.

Om 6:e biten innehåller en etta så hoppar datorn till rad 300.

Det sista sättet används mycket när man skall styra olika saker .

OR är nästa och den används på liknande sätt.

IF A=6 OR B=7 THEN 200.

Nu testar datorn om A eller B uppfyller villkoret det behövs bara att ett av värdena är riktiga. Är båda riktiga så hoppar den också till 400.

I andra fallet används OR när man vill sätta en etta i en viss adress.

POKE 32500, (PEEK (32500 OR 64).

Om inte biten 64 är tänd så blir den det med ovanstående. Vill man släcka en viss bit i en adress så använder man AND. Nu skall vi släcka bit 64 och då tar vi och gör på följande sätt.

POKE 32500, (PEEK (32500 AND (255-64).

255 är det tal som man får när alla bitarna i en byte är tända.

Detta system med AND och OR måste man kunna för att styra olika saker med hjälp av datorn.

Den tredje heter NOT och betyder icke och skrives på följande sätt:

IF NOT A=5 THEN 600.

Lägg märke till att NOT står framför villkoret. Datorn kommer nu att hoppa till rad 600 om A inte är 5.

Alla logiska operatorer kan också användas med strängar. Till exempel med NOT kan du testa ett slut:

IF NOT A$ = "SLUT" THEN 400.

Så ända tills strängen A$ blir ordet SLUT så hoppar datorn till en operator som inte finns i alla basic-dialekter är XOR, som betyder EXCLUSIVE-OR eller på svenska EXKLUSIVT-ELLER. Man också översätta det med antingen eller. Den skiljer sig från OR genom att endast ett villkor skall vara uppfyllt.

IF A=5 XOR B=6 THEN 400.

Nu skall antingen A=5 eller B=6, men hoppar inte till 400 om både A=5 och B=6, vilket den gör om där hade stått OR i stället för XOR. Vad gör man om man inte har xor. Då får man skriva på följande sätt:

IF A=5 AND NOT B=6 OR NOT A=5 AND B=6 THEN 400.

Ser krångligt ut men tänker du lite så är det hela ganska logiskt. Du som läst lite digitalteknik ser att de här logiska operatorerna stämmer helt överens med digitaltekniken.

Det lilla programmet i figur 1. Kan du använda för att testa om din dator har de olika operatorerna. Om du inte har XOR så kommer du att få "SYNTAX ERROR" i rad 60. Stryk då rad 60.

Nu till en läsares fråga om krypteringsprogram. Det är naturligtvis spännande att skicka meddelande som inte andra kan läsa. De flesta krypteringsprogram är mycket enkla och lätta att knäcka. Ett bra krypteringsprogram skall inte ge samma tecken för samma bokstav. Om jag skall kryptera ordet TITTA så innehåller ordet tre T och får man ordet ZNZZF, så behöver man inte mycket hjälp för att komma fram till att det är fem bokstäver man flyttat fram. Om du däremot ser ordet VNEÖS så är det omöjligt att komma fram till att det är ordet TITTA krypterat. I figur 2 så hittar du detta krypteringsprogram, som ger ordet VNEÖS om du använder CHRISTIAN som lösenord.

Lösenord

Programmet frågar först efter ett lösenord och sedan efter texten, som skall krypteras. Kom ihåg att du kan inte kryptera mer än 255 tecken åt gången, kanske mindre du får rådfråga din instruktionsbok om hur långa strängar du kan skriva i en sträng.

I programmet läggs lösenordet i variabeln 0$ och texten, som skall krypteras läggs i variabeln W$. Sedan används lösenordet för att få fram ett siffervärde i rad 1050, som minskas med 65 för att få alfabetiska tecken. Genom att i rad 1060 göra en XOR så får jag i rad 1070 i CHR$(P) fram det nya tecknet, som blir olika även om i ursprungstexten har samma tecken. Bestäm ett lösenord med din kompis, så kan ni skicka meddelande, som ingen kan knäcka. Om du har XOR på din skrivare skriver du i rad 1060 på följande sätt: 1060 P = N XOR B

```
10 PRINT CHR¤(26)
20 INPUT "LÖSENORD";Q¤
30 INPUT "ORDET SOM SKALL KRYPTERAS";W¤
40 GOSUB 1000
60 PRINT W¤
70 PRINT" DEN KRYPTERADE TEXTEN BLIR"
80 PRINT C¤
90 END
1000 Q=LEN(Q¤)
1010 A=1
1020 C¤=""
1030 FOR I=1 TO LEN(W¤)
1040 N=ASC(MID¤(W¤,I,1))
1050 B=ASC(MID¤(Q¤,A,1))-65
1060 P=N AND NOT B OR NOT N AND B
1070 C¤=C¤+CHR¤(P)
1080 IF A=Q THEN A=0
1090 A=A+1
1100 NEXT I
1110 RETURN
```

● Figur 2.

```
10 INPUT A
20 INPUT B
30 IF A=5 OR B=6 THEN PRINT "OR"
40 IF A=5 AND B=6 THEN PRINT "AND"
50 IF NOT A=5 THEN PRINT "NOT"
60 IF A=5 XOR B=6 THEN PRINT"XOR"
70 IF A=5 AND NOT B=6 OR NOT A=5 AND B=6
THEN PRINT" XOR "
```

● Figur 1.

Råd och tips om datordialekter



Hem DATORER

Av Christian Lindeberg

Den här veckan skall jag ta upp en del frågor om programmering. Det är många basicdialekter på våra olika datorer och det händer ofta att man träffar på uttryck, som man inte vet hur man skall översätta till den basic man själv använder. Jag skall ta upp en del sådana uttryck och även visa att det kan vara betydelse att man tänker på hur lång tid ett program skall ta för att uföras. Inom vissa gränser kan man själv bestämma att det skall gå lite fortare.

IF-THEN-ELSE är ett uttryck, som inte alla datorer har. I en del basicdialekter kan du skriva till exempel:

10 IF A=5 THEN PRINT A ELSE PRINT "INTE 5"

Ordet ELSE fungerar som "I ANNAT FALL SÅ". Om din dator inte har ELSE så kan du simulera det genom att använda ON-GOTO. ON-GOTO fungerar på följande sätt: om du skriver ON A GOTO 100, 200, 300 så kommer datorn att hoppa till rad 100 om A=1, till 200 om A=2 o s v. Detta kan man använda plus att man vet att datorn ger värdet 0 om något är falskt och ger siffran -1 om något är sant. Vi kombinerar nu dessa två saker så kan vi skriva om IF-THEN-ELSE på följande sätt:

```
10 FOR A=1 TO 6
20 ON -(A=5) GOTO 30: PRINT "INTE 5": GOTO 40
30 PRINT A
40 NEXT A
```

Om A=5 är sant så hoppar datorn till rad 30 "I ANNAT FALL" hoppar den till rad 40.

Lång tid

Ett annat problem när man kör matematikprogram är att det kan ta väldigt lång tid. Detta beror på att man inte har deklarerat sina variabler i början av sitt program. Normalt är detta något man inte behöver bry sig om, men om man använder datorn till beräkningar får det en mycket stor betydelse. Jag skall visa med några enkla exempel.

```
10 A=0:B=0:C=0:E=0-:F=0:G=0:H=0:I=0
20 J=0:K=0:L=0:M=0-:N=0:O=0:P=0:Q=0-:R=0
30 S=0:T=0:U=0:V=0-:W=0:X=0:Y=0:Z=0
40 TI#="000000"
50 FOR X=1 TO 10000
60 Y=Y+Z
70 NEXT X
80 A1=TI: PRINT A1/60; "SEKUNDER"
```

Programmet är skrivet för att passa VIC-datorerna, men ta bort rad 40 och 80 och mät tiden med din egen klocka, så går det lika bra. Det bör ta ungefär 7 sekunder att köra programmet. Du ser att jag har använt variablerna X, Y och Z. Dessa variabler är deklarerade i slutet på rad 30. Ändra nu programmet på följande sätt:

```
50 FOR A=1 TO 1000
60 B=B+C
70 NEXT A
```

Värdet "noll"

Kör nu programmet en gång till och notera tiden. Den bör ha blivit ungefär 4 sekunder, alltså nästan hälften av vad det första programmet gav. Varför? När datorn letar efter variabler så börjar den i början av variabellistan och fortsätter tills den hittar sin variabel. I första programmet låg variablerna sist, men i det andra var det de första variablerna. Detta gjorde att programmet gick nästan dubbelt så fort. Om man inte deklarerar sina variabler i början av ett program så skapas de efter hand och kan då göra att ett program tar mycket längre tid än vad det normalt skulle behöva göra. Så kom ihåg att när du gör program med många beräkningar så ge alla dina variabler värdet noll i början av ditt program, så gör du det också snabbare. En annan sak att tänka på är att hur man skriver sina matematiska uttryck. Om du använder sista programmet och ändrar i rad 60 till följande:

60 B=75★75

Kör programmet och ta tiden, det bör bli ungefär 7 till 8 sekunder. Ändra nu rad 60 till följande:

60 B=75Ü2

I rad 60 har jag använt tecknet upphöjt till, som kan vara en uppåtriktad pil eller ett tyskt y. Kör nu programmet och du bör få en tid på 55 till 60 sekunder. Använd inte tecknet för upphöjt till i dina program för att det slöar ner dem katastrofalt.

Du som har en COMMODORE 128 kan använda funktionen FAST för att snabba upp programmen. Vet du att du kan använda denna funktion också i 64-läge? Med en 128 kan du snabba upp dina beräkningar med dubbel hastighet i 64-läge på följande sätt:

Skriv POKE 53296, 1 för att sätta på FAST-läge. Nu kommer skärmen att se konstig ut, så att du kan inte använda detta läge för att skriva ut något på skärmen, men för att snabba upp beräkningar kan du göra på följande sätt: Använd programmet vi hade tidigare med upphöjt till. Programmet, som tog 55 sekunder. Gör du följande tillägg:

```
35 POKE 53296,1: POKE 53265,11
75 POKE 53296,0: POKE 53265,27
```

Rad 35 slår över till FAST stänger bildskärmen.

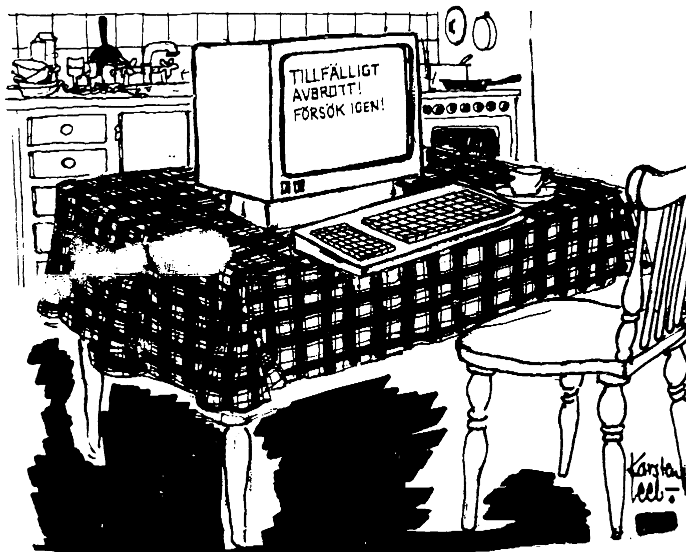
Rad 75 går tillbaka till normalläge.

Adressen 53296 sätter alltså på FAST med en ETTA och stänger med en NOLLA.

Adressen 53265 stänger av skärmen med talet 11 och återgår till normalläge med talet 27.

Utländsk tidning

Det går alltså att komma åt funktionerna på 128:an även i 64-läge om du har möjlighet med 80 tecken på din skärm så kan du också få 80 tecken i 64-läge. Har du till exempel SPEEDSCRIPT version 3.0 eller högre så kan du köra detta med 80-tecken skärm genom att komplettera ditt ordbehandlingsprogram. Om du är intresserad så köp Computers Gazette för juni månad. Tidningen kommer ut på svenska marknaden i början eller i mitten av juni månad. Jag håller på att undersöka marknaden för CP/M program till 128:an. Det är många som skrivit och frågat och jag hoppas att jag har fått ihop material innan sommarlovet. Om du har någon information om CP/M till 128:an så skriv snabbt.



● *Ge inte upp! Alla datorer har uttryck som motsvarar en annan dators.*

Lagring av tabeller i "hopknåpat" program

Det närmar sig slutet av terminen och slutet på den här omgången artiklar. Nästa lördag är sista gången den här terminen. De av er som får problem i sommar med sina program kan skriva direkt till mig. För dem som inte har listat ut min adress så lämnar jag den i slutet av artikeln. Dagens artikel kommer att bli kort, men med en desto större lista på ett program.



Hem DATORER

Av Christian Lindeberg

Innan programmet skall jag tala om att det andra numret av "Svenska HEMDATOR hacking" har kommit ut. Numret består av en del hopskrap från andra datortidningar. Det är definitivt inte tid att börja prenumerera än. Fortsätter den i samma stil så kommer den att dö ut, som de andra en hel del av oss har förlorat pengar på.

Ny tidning

Men det har kommit en ny tidning för C-64/C-128 ägare som heter "Datormagasin C64/C128" — kostar 12 kronor. Tidningen är i kvällstidningsformat och tryckt på vanligt tidningspapper. Tryckningsmässigt av lägre kvalitet än den ovanstående, men innehållsmässigt flera nivåer högre. En tidning, som är värd sitt pris och som jag hoppas överlever, vilket kommer att bero på oss. Programmet ni ser i dagens artikel har uppstått från en mängd brev, som jag fått efter mitt förra tabellprogram i vintras. Ola Axelsson i Hässleholm var den, som fick mig att sätta mig ner och "knåpa ihop ett program". Ola har en VIC-2Ø och är en stor optimist. Först tänkte jag att det är omöjligt att klara det på en 2Ø:a, men jag har gjort ett försök, så att det kan gå in på en VIC med bara 3,5 KB.

Jag har skrivit programmet för floppydisk, men du skall få ändringar här om du har en bandspelare.
Rad 4ØØØ ändras till
4ØØØ OPEN 2,1,Ø,P§
Rad 8ØØØ ändras till
8ØØØ OPEN 2,1,1P§

Om du använder bandspelare så använd ett separat band för lagring av tabellen.

Lagra tabeller

Jag tror att ni har förstått att programmet handlar om att lagra en tabell t ex i fotboll. Datorn tar hand om sortering när man lagrat nya uppgifter. Utskrift av tabell är gjort med tanke på den begränsade skärmen, som VIC:en har. Ju större skärmbredd du har desto fler saker kan du skriva ut i tabellutskriften.

Du, som har bandspelare, tänk på att spola bandet till rätt ställe, innan datorn börjar spela in eller av bandet.

Du som inte har en VIC kan ändra i rad 4ØØØ och 8ØØØ, så att det stämmer med din instruktionsbok.

Du ser att jag verkligen "knåpat ihop programmet" därför alla dessa GOSUB-satser, som kom till efter hand som jag skrev programmet och kom på saker jag hade glömt. Programmet är inte någon skönhet vad det gäller utskriften på skärm, men du kan själv snygga till det när det regnar i sommar.

Här kommer min adress om du vill skriva i sommar:
CHRISTIAN LINDEBERG
Hasselviksvägen 24
236 00 Höllviken
Nästa vecka sista innan sommarlovet.

```
10 DIM L¤(14,7)
15 GOSUB 7000
20 PRINT CHR¤(26)
30 PRINT"   MENY"
40 PRINT"1. NYA RESULTAT"
50 PRINT"2. VISA TABELL"
60 PRINT"3. STATUS"
80 PRINT"4. HÄMTA TABELL"
90 PRINT"5. GöRA NY TABELL"
100 PRINT"6. AVSLUTA"
110 INPUT "   VAD VILL DU GöRA";A
120 ON A GOSUB 1000,2000,3000,4000,5000,6000
130 GOTO 20
1000 PRINT CHR¤(26)
1010 PRINT"SKRIV V=VINST, O=OAVGJORT, F=FöRLUST OCH E=EJ SPELAT"
1020 FOR I=1 TO N
1040 PRINT"LAG ";L¤(I,1);
1050 INPUT "V, O, F ELLER E";M¤
1060 IF M¤="V" THEN GOSUB 1500
1070 IF M¤="O" THEN GOSUB 1600
1080 IF M¤="F" THEN GOSUB 1700
1090 IF M¤="E" THEN 1130
1100 INPUT "MALSKILLNAD";T
1110 B=VAL(L¤(I,6))+T:L¤(I,6)=STR¤(B)
1120 B=VAL(L¤(I,2))+1:L¤(I,2)=STR¤(B)
1130 NEXT I
1140 GOSUB 9000
1150 GOSUB 8000
1160 RETURN
1500 B=VAL(L¤(I,3))+1
1520 L¤(I,3)=STR¤(B)
1530 B=VAL(L¤(I,7))+2
1550 L¤(I,7)=STR¤(B)
1560 RETURN
1600 B=VAL(L¤(I,4))+1
1620 L¤(I,4)=STR¤(B)
1630 B=VAL(L¤(I,7))+1
1650 L¤(I,7)=STR¤(B)
1660 RETURN
1700 B=VAL(L¤(I,5))+1
1710 L¤(I,5)=STR¤(B)
1720 RETURN
2000 PRINT CHR¤(26)
2010 PRINT"LAG";TAB(10);"M-SKILL.";TAB(15);"POÄNG"
2020 FOR I=1 TO N
2030 PRINT L¤(I,1);TAB(11);L¤(I,6);TAB(16);L¤(I,7)
2040 NEXT I
2045 INPUT "TRYCK PÅ RETURN";QQ¤
2050 RETURN
3000 PRINT CHR¤(26)
3010 FOR I=1 TO N
3040 PRINT"LAG";L¤(I,1)
3050 PRINT"ANTAL MATCHER";L¤(I,2)
3060 PRINT"VINSTER";L¤(I,3)
3070 PRINT"OAVGJORDA";L¤(I,4)
3080 PRINT"FöRLUSTER";L¤(I,5)
3090 PRINT"MALSKILLNAD";L¤(I,6)
3100 PRINT"POÄNG";L¤(I,7)
3110 PRINT:PRINT
3120 INPUT "TRYCK PÅ RETURN";QQ¤
3130 NEXT I
3140 RETURN
4000 OPEN 15,8,15,"I":CLOSE 15:OPEN 2,8,2,P¤+"S,R"
4010 INPUT#2,N
4020 FOR I=1 TO N
4030 FOR J=1 TO 7
4040 INPUT#2,L¤(I,J)
4050 NEXT J
4060 NEXT I
4070 CLOSE 2
4080 RETURN
5000 PRINT CHR¤(26)
5005 INPUT "HUR MANGA LAG";N
5010 FOR I=1 TO N
5020 INPUT"LAG";L¤(I,1)
5030 INPUT"ANTAL MATCHER";L¤(I,2)
5040 INPUT"VINSTER";L¤(I,3)
5050 INPUT"OAVGJORDA";L¤(I,4)
5060 INPUT"FöRLUSTER";L¤(I,5)
5070 INPUT"MALSKILLNAD";L¤(I,6)
5080 INPUT"POÄNG";L¤(I,7)
5090 NEXT I
5100 GOSUB 8000
5110 RETURN
6000 END
7000 PRINT CHR¤(26)
7010 PRINT"VAD HETER TABELLEN"
7020 INPUT P¤
7030 RETURN
8000 OPEN 15,8,15,"I":CLOSE 15:OPEN 2,8,2,"£:",+P¤+"S,W"
8010 PRINT#2,N
8020 FOR I=1 TO N
8030 FOR J=1 TO 7
8050 PRINT#2,L¤(I,J)
8060 NEXT J
8070 NEXT I
8080 CLOSE 2
8090 RETURN
9000 FOR I=1 TO N-1
9010 FOR J=I+1 TO N
9020 IF L¤(I,7)>L¤(J,7) THEN 9080
9030 FOR K=1 TO 7
9040 X¤=L¤(I,K)
9050 L¤(I,K)=L¤(J,K)
9060 L¤(J,K)=X¤
9070 NEXT K
9080 NEXT J
9090 NEXT I
9100 RETURN
```

Olika databaser över hela landet



Av Christian Lindeberg

Detta är den sista artikeln före sommaren och jag skall försöka städa ut de frågor, som är kvar från den här terminen. Den första frågan gäller datakommunikation.

Vad finns det för databaser, som man kan komma i kontakt med och vad kostar det?

Det finns en del olika typer av baser beroende på vilken dator man använder. ABC-datorerna har ett par databaser i den närmaste är i Halmstad 035/110771 och heter ABC-banken och kör bara med 300/300 modem. Atari har vad jag vet bara en bas och den ligger i Stockholm och har nummer 08/7710280 och heter "The Strapper Base". VIC-datorerna har flera baser, den närmaste är i Vänersborg och har nummer 031/219424, men kan bara köras med C/G Term och 300/300 modem. Sedan finns det en del Fido-baser, som är så kallade "public domain"-baser, där man kan få hämta program gratis. Det finns en Fido-bas i Malmö, som även har program till VIC-datorer. Fido är ett så kallat bulletinboard system (BBS) främst med tanke på MS-DOS datorer men har även till VIC och en del programvara i CP/M och en del kommer till ABC 80/800. I Fido kommer man också att ha datormöte, så att man kan diskutera olika datorproblem. Fido kan hantera 300/300, 75/1200 och 1200/1200 full duplex. Telefonnumret är 040/549189. Ställ ditt program på 09 databitar och en stoppbit om du skall köra mot Fido.

Caves

Nästa fråga gäller vad för några porogram man kan köra på Commodores 128 i CP/M-läge.

Alla program, som är skrivna i CP/M för Kaypro eller Osborne går att köra och läsa i 128:an. Det finns många program, allt från Wordstar, Multiplan till många språk Basic och Turbo-pascal. Det går även att få tag i det klassiska adventure-programmet Grottor eller Caves, som det heter på engelska. Caves är uppdelat på flera bitar och tar totalt cirka 200 KB, ett stort program.

Stön

Ola i Malmö har problem med sin nya 128:a med 1571:ans diskdrive. Problemet består av att när man växlar läge mellan 128 och 64 läge, så står diskdriven och stönar och låter innan den hittar rätt format.

```
READY.

10 PRINTCHR¤(147)
20 FOR X=0 TO 50
30 IF PEEK(X+1344)=43 THEN POKE X+1344,42
40 IF PEEK(1863-X)=43 THEN POKE 1863-X,42
50 NEXT X
60 SS¤="                                      "
70 B¤=SS¤+"  CHRISTIAN VISAR"
80 B¤=B¤+"- SKROLLNING - PA VIC-DATORER"
90 B¤=B¤+" DETTA PROGRAM KAN DU"
100 B¤=B¤+" ANVÄNDA I DIN VIC-DATOR"
110 B¤=B¤+" DET ÄR BARA ATT SKRIVA SIN TEXT"
120 B¤=B¤+" OCH INTE GLÖMMA ETT MELLANSLAG"
130 B¤=B¤+" INNAN MAN BÖRJAR SIN TEXT "
140 B¤=B¤+"OCH ETT I SLUTET  "
150 D=0
160 S=1:IF D=2 THEN 20
200 PRINTCHR¤(19);MID¤(B¤,S,40)
210 FOR I=0 TO 90:NEXT I:S=S+1
220 IF S=LEN(B¤) THEN D=2: GOTO 160
230 GOTO 200

READY.
```

Det är på det viset att 1571 uppför sig som en enkelsidig drive, alltså som en 1541 i 64-läge och som en dubbelsidig drive i 128-läge. För att komma ifrån detta problem, så gör på följande sätt. Om du skall köra i 64-läge, så skriver du OPEN 15,8,15, "UO>M0": CLOSE15 och i 128-läge skriver du OPEN 15,8,15,"U0>M1": CLOSE15.

PRINT USING är en basicfunktion, som inte finns i många basicdialekter. Funktionen tillåter att man skriver med ett visst antal siffror i decimalerna. Ett problem är att skall man skriva ut priser, så skriver datorn 2.5 om priset är 2.50 detta kan man avhjälpa med PRING USING, där man kan bestämma antalet decimaler så att datorn skriver 2.50. Har man inte PRINT USING, så kan man enkelt snickra ihop en egen liten subrutin. Kör följande program: 10 INPUT "ETT TAL";A
20 T=A:GOSUB 10000:PRINT T$
30 GOTO 10
10000 F$=STR$(INT(T)):S$="."
10010 IF T<>INT(T) THEN S$=S$+MID$(STR$(100+(-T★100) — INT(T)★100+0.001),3,2)
10020 S$+LEFT$(S$+"00",3):T$=-RIGHT$(" "+F$+S$,9)
10030 RETURN

Det är alltså subrutinen i raderna 10000 till 10030 som gör det hela. Denna rutin kan du använda i dina program.

Monitor

Det har kommit många frågor om RGB och RGBI till 128:an.

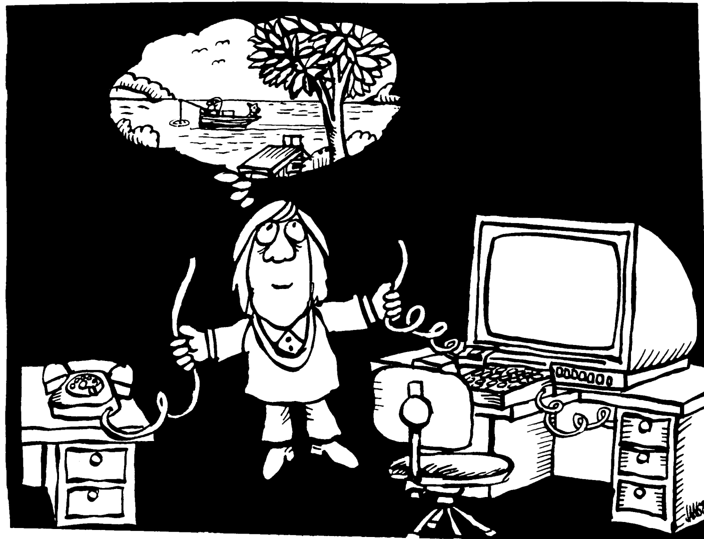
Till 128:an måste det vara en monitor med digatal RGB eller RGBI. Det går alltså inte med en TV, som har analog RGB utan att man har ytterligare en burk emellan TV och dator. Men om man utnyttjar 80-tecken på 128:an så använd en RF-modulator och en vanlig TV, så blir det billigare.

I Allt om Elektronik nummer 5/1985 fanns en byggsats på en RF-modulator, som man också kan köpa i byggsats från Allt om Elektronik.

TIll dig, som har en 128:a, skriv så här på din dator SYS 32800,123,45,6 och tryck på RETURN.

Litet program

Till slut har jag ett litet program, som skrollar fram text på din 64:a. Tekniken kan användas på alla datorer. Tänk på att du måste ha mellanslag i början av varje textrad annars skriver datorn ut meddelandet sammanhängande utan mellanslag. Det är också viktigt att man har ett mellanslag i slutet för annars kommer sista bokstaven att repeteras över hela skärmen. Till sist vill jag önska alla en riktigt trevlig och varm sommar.



● *I dag drar Christian Lindeberg ur sladden till sina hemdatorer för den här terminen. Han önskar alla en trevlig sommar och på återseende till hösten.*

Ett vad som du aldrig förlorar

Marknadskriget mellan Atari och Commodore startade i förra veckan. Amiga 500 kostar 6.500 kronor mot Atari 520 ST, som kostar 4.500 kronor. Den stora frågan är nu om Amiga 500 är 2.000 kronor bättre än Atari ST. Jag hoppas att kunna prova båda datorerna under sommaren och försöka komma med ett svar så småningom.



Hem DATORER

Av Christian Lindeberg

Det kom ett brev från Mats S i Malmö, som undrade om man kunde göra en enkel enarmad bandit i basic. Jag har skrivit ihop ett litet program i basic, som använder slumptalsgeneratorn för att ta ut tre tal till den enarmade banditen. Dessa tre tal kan man naturligtvis också använda för att plocka ut olika symboler från 64:ans teckenuppsättning. Programmet jag har gjort använder bara siffror, men du kan själv komplettera med symboler, som hjärter, ruter eller andra symboler.

Undvik Å-Ä-Ö

I programmet har jag använt mig av hela ord för variablerna. När man gör stora program skall man alltid använda sådana namn på variablerna att de är lätta att känna igen. Här har för de tre talen använt mig av variablerna ETT, TVA och TRE. Nu undrar nog någon varför använder han ordet TVA och inte TVÅ, som det heter på riktigt. Anledningen är att på många datorer fungerar inte variabel namn om man använder ÅÄÖ, eftersom dessa inte är naturliga från början i datorn. Ni vet att det engelska alfabetet slutar på Z och därefter har man vad som kallas för hakparenteser och ett pundtecken. Dessa tecken har fått lämna plats för de svenska tecken för ÅÄÖ. I datorspråkets uppbyggnad har man använt de här symbolerna speciellt hakparentes till att betyda vissa saker när de använts. Datorn ser inte om det är en hakparentes eller ett Å utan avläser bara att den har en viss ASCII-kod, som för datorn betyder något speciellt. För att undvika problem, så använd aldrig ÅÄÖ i namn på varialber.

Satsa mera!

Låt oss titta på programmet!

Rad 10 PRINT CHR$(147) är koden för att tömma skärmen på Commodores datorer. I GW-basic skall man skriva CLS, vilket är en förkortning av "Clear Screen", på ABC-datorerna skriver man PRINT CHR$(12). Det finns fler varianter men de har samma funktion att tömma skärmen innan programmet börjar skriva något på skärmen.

Rad 20 är utskrift av programnamnet.

Rad 30 är en vänteloop. Det händer ingenting i loopen, men datorn väntar här i lite mer än en sekund.

Raderna 50 och 60 tar hand om ditt vad, alltså hur mycket du satsar. Du ser att jag har också kallat variablen för VAD.

Rad 70 räknar ihop hur mycket du har satsat i variabeln PENG.

Rad 80 ger två tomma rader.

Rad 90 kontrollerar hur mycket du har satsat. Har du satsat mindre än 200 så vill datorn att du satsar mer, som den också talar om i rad 100.

Slumptal

Rad 110 måste med så att datorn räknar ifrån dina felsatsade pengar.

Rad 120 hoppar till rad 50 så du kan öka din insats.

Rad 130 till 150 talar om hur mycket du får tillbaka om du har två eller tre lika siffror.

Rad 160 Här väntar datorn tills du har tryckt på RETURN. Detta är ett bra sätt när man vill att datorn skall vänta tills man är klar.

Rad 170 och 250 här har jag gjort en lopp som snurrar ett slumpartat antal gånger. Loopen går från 3 och i bästa fall upp till 13. Lägg märke till PRINT CHR$(147) efter loopen. Genom att tömma skärmen var gång det blir en ny slumptalserie så kommer talen att blinka fram på skärmen och bara de tre sista kommer att vara kvar på skärmen.

Rad 180 till 210 här tar datorn ut slumptal mellan 0 och 8. Dessa slumptal lägger datorn i variablerna ETT, TVA och TRE.

Rad 220 skriver ut slumptalen på skärmen med hjälp av TAB-funktionen.

Rad 230 en liten vänteloop så att man hinner se vad datorn skriver ut. Testa med att ändra värdet 250 och se vad som händer. Från raderna 270 till 450 så testar datorn vilka siffror som kom upp i variabla ETT, TVA och TRE. Datorn jämför parvis om de olika variablerna är lika. Om ETT är lika med TVA så testar den sedan om TVA är lika med TRE och är det fallet så är alla tre siffrorna samma. Det kan också skrivas med hjälp av AND och OR men jag tycker att detta sätt blir lite mer överskådligt.

Raderna 460 till 500 skriver ut hur mycket du vunnit eller förlorat och frågar om du vill fortsätta.

Detta är en typ av program där du aldrig behöver förlora. Slå vad med din kompis (om han inte har läst den här artikeln) att du kan alltid vinna pengar av datorn och att datorn alltid förlorar.

Nu är det din tur — du satsar först 300 om du vinner nu var det lätt, men vi säger att du förlorar. Nu satsar du dubbelt så mycket, alltså 600. Förlorar du igen så fördubblar du igen till 1.200. Du håller på ända tills du vinner och då har du vinst mot datorn. Du kan alltså aldrig förlora med detta system.

```
10 PRINT CHR$(147)
20 PRINT:PRINT"DEN ENARMADE BANDITEN"
30 FOR I=1 TO 1000:NEXT I
40 PRINT CHR$(147)
50 PRINT:PRINT"HUR MYCKET SATSAR DU"
60 PRINT:PRINT:INPUT "SKRIV ETT TAL OCH TRYCK PÅ
                                      RETURN";VAD
70 PENG=PENG+VAD
80 PRINT:PRINT
90 IF VAD>200 THEN 130
100 PRINT:PRINT"MER VILL JAG HA"
110 PENG=PENG-VAD
120 GOTO 50
130 PRINT:PRINT"TRE AV SAMMA SIFFRA GER ";3*VAD
140 PRINT:PRINT"TVÅ AV SAMMA SIFFRA GER ";2*VAD
150 PRINT:PRINT
160 INPUT "TRYCK PÅ RETURN";TRYCK$
170 FOR K=1 TO INT(RND*10+3):PRINT CHR$(147)
180 PRINT:PRINT:PRINT:PRINT
190 ETT=INT(RND*9)
200 TVA=INT(RND*9)
210 TRE=INT(RND*9)
220 PRINT TAB(15);ETT;TAB(25);TVA;TAB(35);TRE
230 FOR J=1 TO 250:NEXT
240 PRINT:PRINT
250 NEXT
260 INPUT "TRYCK PÅ RETURN";TRYCK$
270 IF ETT=TVA THEN 290
280 GOTO 370
290 IF TVA=TRE THEN 330
300 PRINT:PRINT"DU VANN";2*VAD
310 PENG=PENG-2*VAD
320 GOTO 460
330 PRINT:PRINT"DU VANN";
340 PRINT 3*VAD
350 PENG=PENG-3*VAD
360 GOTO 460
370 IF ETT=TRE THEN 420
380 IF TVA=TRE THEN 420
390 PRINT:PRINT"DU VANN INGET"
400 FOR I=1 TO 1000:NEXT
410 GOTO 460
420 PRINT:PRINT"DU VANN";2*VAD
430 PENG=PENG-2*VAD
440 PRINT:PRINT
450 PRINT:PRINT
460 PRINT:PRINT"VILL DU FORSATTA!"
470 PRINT:PRINT:IF PENG>0 THEN PRINT "DU HAR
                        HITILLS FöRLORAT ";PENG
480 PRINT:PRINT:IF PENG<0 THEN PRINT "DU HAR
                          HITILLS VUNNIT";-PENG
490 PRINT:PRINT
500 INPUT"SKRIV J ELLER N OCH TRYCK PÅ RETURN"
                                          ;SVAR$
510 IF SVAR$="J" THEN 40
520 IF SVAR$="N" THEN END
530 GOTO 460
```

Problem med sorteringsprogrammen?

Inget fel på datorn — den arbetar!

Nu är sommarlovet slut och höstterminen har börjat. Jag hoppas att vädret blir bättre nu än vad det har varit i sommar. Allt det här regnandet har gett många tillfälle att sitta framför datorn, men jag hade gärna bytt datortid mot lite solsken och bad.



Hem DATORER

Av Christian Lindeberg

Det har kommit många brev i sommar och jag skall besvara en del av dem. Först till breven och jag börjar med ett från Fredrik i Landskrona. Fredrik undrar om det finns någon möjlighet att koppla modem till en ZX81 för att använda den till kommunikation. Jag har sett burkar att koppla ihop med en ZX81 för att sedan kunna koppla in ett modem. Jag vet inte vad dessa burkar kostar men ta kontakt med Beckman Innovation i Stockholm, adressen är Box 1007, 122 21 Enskede, telefon 08-39 04 00.

Police Academy

Jag har fått brev ifrån Pål och Henrik i Lund. De har ett spel som heter "POLICE ACADEMY". Vad jag vet finns det inte något spel ute i affärerna med det namnet. Däremot har jag hört talas om ett program, som heter "POLICE ACADEMY" och som är ett skämt för vad du än skriver så försvinner allt på skärmen. Det verkar som om ni har blivit utsatta för ett skämt.

Pål har skrivit ytterligare ett brev angående ett program som jag hade den 2 maj. Tyvärr har jag inte den listning, som fanns i tidningen utan en annan. Problemet är att jag inte vet hur din rad 500 ser ut. Ditt problem hänger kanske samman med vad "Mr X u Y" från Eslöv skrev om. Den 30/5 skrev jag att ordet SALDO inte innehöll något reserverat ord. Jag hade fel för i ordet SALDO finns ordet DO, som är ett reserverat ord hos en del datorer.

Datorn arbetar

En fråga som jag har fått i ett flertal brev är: Varför hänger min dator sig när jag kör mitt sorteringsprogram? Det är inget fel på programmet, men datorn hänger sig och jag måste stänga datorn. Svaret är att datorn inte hänger sig utan att den arbetar. Vad det hela handlar om är "garbage collection". Skriv följande program för att se effekten av "garbage collection":

```
10 X$="A":J=0:S=0
20 DIM A$(9000)
30 FOR J=0 TO 9000
40 PRINT J:T=TI
50 A$(J)="A"+"":A$(9000-J)=
   "B"+""
60 A$(J)="C"+"":A$(9000-J)=
   "D"+""
70 S=TI-T:PRINT S,
80 IF S>3600 THEN STOP
90 NEXT J
```

Programmet kommer att skapa en mängd strängar och efter cirka fyra minuter kommer datorn att stanna och börja med sin "garbage collection". Den kan i värsta fall ta upp till 61 minuter. Under denna tid är datorn helt låst och kan inte göra något. Vad som har hänt är att datorn inte har något fritt minnesutrymme kvar för dynamiska strängar.

Högsta adressen

Man skiljer på två olika typer av variabler. De variabler, som arbetar med siffror kallas numeriska variabler och de, som tar hand om bokstäver och tecken kallas för strängar. Ovanför basicprogrammet lägger datorn alla numeriska variabler och dessa fylls på uppåt och det finns en pekare, som håller reda på den högsta adressen.

Strängarna däremot startar i toppen på minnet och går neråt, och även här finns en pekare, som håller reda på den lägsta adressen. När pekaren från variablerna kommer uppåt och pekaren från strängarna kommer neråt så minskar det fria minnesutrymmet. Om pekarna träffar på varandra under körning så inträffar en "garbage collection". Om pekarna träffar på varandra omedelbart efter det att du skriver RUN, så är programmet för stort för din dators minne och då får du meddelandet "OUT OF MEMORY ERROR".

Statistiska strängar

Jag skall förklara vad som händer vid en "garbage collection". Först måste vi skilja på statiska och dynamiska strängar. En statisk sträng hittar du i rad 10 i programmet X$="A". När man skapar en sträng så lägger datorn upp en variabel beskrivning (på engelska "variable descriptor"), som innehåller tre olika delar:

1. Variabelnamnet
2. Strängens längd
3. En pekare till den adress där variabeln börjar

Man kan se det på följande sätt:
Namn: X$
Längd: 1 byte
Adress: 2057 ($0809, den hexadecimala adressen)

Det betyder att bokstaven A ligger i adress 2057. Detta är en statistisk variabel, när den ligger i basicprogrammet. Om istället X$="CHRIS"+"TIAN", så måste datorn först bilda en sträng "CHRISTIAN", som den sedan lägger högst upp minnet med en variabel beskrivning på följande sätt:
Namn: X$
Längd: 9 bytes
Adress: 40949 ($9FF5)

Detta är en dynamisk sträng och det är dessa, som skapar våra problem.

Variabelbeskrivning

INPUT och GET skapar alltid dynamiska strängar. Om vi i ett program har till exempel en rad INPUT A$. Första gången datorn kommer dit, så blir A$="SVEN". Datorn skapar då en variabelbeskrivning på var "SVEN" ligger i minnet. Nu kommer ett nytt namn i INPUT A$ och A$="STEFAN". Vad händer? Variabelbeskrivningen kommer då att ha kvar namn=A$, men längden ändras till 6 bytes och pekaren till adressen var den ligger ändras, därför att datorn lägger inte namnet STEFAN över namnet SVEN utan läger namnet STEFAN under namnet SVEN. Skillnaden blir att det finns ingen pekare till namnet SVEN, men det ligger fortfarande kvar och tar upp plats i minnet.

Rensa minnet

Efter hand som A$ byter innehåll så flyttas pekarna neråt, för att de gamla namnen ligger kvar. När pekaren träffar på pekaren för de numeriska variablerna, så märks detta genom att datorn upphör med sin verksamhet och börjar med sin "garbage collection". Vad datorn gör är att titta igenom alla variabelbeskrivningar och spara dem, som är aktiva och slänga bort dem, som inte skall användas. Om man har otur kan datorn vara inaktiv upp till EN TIMME, för så lång tid kan det ta för att rensa upp minnet.



SKRIV GÄRNA

till Christian Lindeberg om allt som har med hemdatorer att göra: program, märken, problem, frågor etc. Adressen är:
Hemdatorer, Arbetet, Box 125, 201 21 MALMÖ.

Reducera V 65-systemet

Hem DATORER

Av Christian Lindeberg

Den här veckan och de följande skall jag ge efter för alla de som har skrivit och undrat hur man gör ett program för att reducera ett V 65-system. Det är svårt att få med alla varianter, men jag har gjort ett ganska stort program, som jag måste dela upp på tre veckor.

Tekniken i programmet är att man i fyra lopp roterar hästarna enligt ett visst system. Man kan rotera 3, 6 eller 9 hästar eller från 2 till 8 hästar matematiskt garderade. Programmet räknar själv ut det totala systemet och talar även om till vilket radantal datorn reducerat systemet och vad det kostar.

De roterande loppen tas ut enligt följande modell. Jag markerar med siffrorna 1, 2 och 3 för att visa principen.

```
1 1 1 2 2 2 3 3 3
1 2 3 1 2 3 1 2 3
2 3 1 3 1 2 1 2 3
3 1 2 2 3 1 1 2 3
```

1 kan betyda häst 1 eller häst 1 och 2 eller häst 1, 2 och 3.

Detta är beroende på om man roterar 3, 6 eller nio hästar.

När du skriver in programmet så tänk på att använda exakt samma radnummer, som jag har gjort för annars kan du inte komplettera med de nästa två veckorna.

Den del av programmet som du får denna vecka tillåter dej att köra med säkra hästar och att matematiskt gardera med 2 hästar. I fyra lopp kan du rotera med sex hästar. Om du skall ha 1 säker i ett lopp så slå in även hästarna 2 och 3 eftersom datorn även tar fram reserver. I fallet med 6 hästar så tar t ex datorn ut häst 1 och 2 och skriver 3 och 4 som reserver. När 3 och 4 tas ut blir 1 och 2 reserver.

Tänk på att CLS är detsamma som PRINT CHR$(147) på Commodoredatorer. Kontrollera med din instruktionsbok vad kommandot för att rensa skärm heter. Om du har en engelsk bok så slå upp på "CLEAR SCREEN". Lycka till med första delen.

```
10 CLS
15 DIM H$(6,15)
20 PRINT:PRINT:PRINT TAB(10);"INMATNING AV HÄSTAR TILL V65"
30 PRINT:PRINT TAB(10);"MATA IN HÄSTENS NUMMER OCH NAMN"
40 PRINT:PRINT TAB(10);"I DE SEX OLIKA LOPPEN"
50 PRINT:PRINT TAB(10);"OCH I DIN RANKING ORDNING"
60 PRINT:PRINT TAB(10);"I VARJE LOPP SLÅR DU IN"
70 PRINT:PRINT TAB(10);"MAX 15 HÄSTAR ELLER MINDRE BEROENDE"
80 PRINT:PRINT TAB(10);"PÅ VILKET SYSTEM DU VILL HA"
90 FOR I=1 TO 6
100 PRINT:PRINT TAB(10);"LOPP NUMMER";I
110 PRINT:PRINT TAB(10);"HUR MÅNGA HÄSTAR";:INPUT N
120 FOR J=1 TO N
130 PRINT TAB(10);"HÄST NUMMER";J;:INPUT H$(I,J)
135 NEXT J:NEXT I
140 CLS:PRINT TAB(10);"REDUCERADE SYSTEM MED FÖLJANDE KOMBINATIONER"
150 PRINT:PRINT TAB(10);"HÄSTARNA KAN SPIKAS GENOM ATT ANGE 1"
160 PRINT:PRINT TAB(10);"3 ST HÄSTAR SOM ROTERAR EN OCH EN HETER 31,32,33 OCH 34"
170 PRINT:PRINT TAB(10);"6 ST HÄSTAR SOM ROTERAR TVÅ OCH TVÅ HETER 61,62,63 OCH 64"
180 PRINT:PRINT TAB(10);"9 ST HÄSTAR SOM ROTERAR TRE OCH TRE HETER 91,92,93 OCH 94"
190 PRINT:PRINT TAB(10);"MATEMATISKA GARDERINGAR SKRIV 2, 3, 4, 5, 6, 7 ELLER 8"
200 PRINT:PRINT TAB(10);"TÄNK PÅ ATT OM DU ANVÄNDER ROTERANDE SYSTEM"
210 PRINT:PRINT TAB(10);"MÅSTE ENTALSSIFFRORNA VARA I FÖLJD 31,62,93,64"
220 D=1:G=1
230 FOR I=1 TO 6
240 PRINT:PRINT TAB(10);"VILKEN KOMBINATION I LOPP";I;:INPUT HS(I)
250 IF HS(I)|10 THEN L=L+1
260 IF LEN(STR$(HS(I)))|2 THEN D=D*VAL(MID$(STR$(HS(I)),2,1))
270 IF LEN(STR$(HS(I)))|2 THEN G=G*(VAL(MID$(STR$(HS(I)),2,1)))/3
280 IF LEN(STR$(HS(I)))=2 THEN D=D*HS(I)
290 IF LEN(STR$(HS(I)))=2 THEN G=G*HS(I)
300 NEXT I
310 PRINT:PRINT TAB(10);"SYSTEMET ÄR PÅ";D" RADER OCH REDUCERAS TILL";G*9"RADER"
315 IF L=6 THEN PRINT:PRINT TAB(10);"OCH KOSTAR"G*.7"KRONOR":GOTO 330
320 PRINT:PRINT TAB(10);"OCH KOSTAR"G*9*.7"KRONOR"
330 PRINT:PRINT TAB(10);:INPUT "ÄR DETTA OK (J/N)";Q$
340 IF Q$="J" OR Q$="j" THEN 360
350 GOTO 140
360 FOR F=1 TO 9
370 CLS:PRINT:PRINT"LAPP NR.";F
380 PRINT
390 FOR I=1 TO 6
400 IF HS(I)=1 THEN GOSUB 660
410 IF HS(I)=2 THEN GOSUB 700
420 IF HS(I)=63 THEN GOSUB 750
430 IF HS(I)=64 THEN GOSUB 1120
440 IF HS(I)=91 THEN GOSUB 1490
450 IF HS(I)=62 THEN GOSUB 1860
460 IF HS(I)=61 THEN GOSUB 2230
470 IF HS(I)=92 THEN GOSUB 2610
480 IF HS(I)=93 THEN GOSUB 2980
490 IF HS(I)=94 THEN GOSUB 3350
500 IF HS(I)=31 THEN GOSUB 3720
510 IF HS(I)=32 THEN GOSUB 4090
520 IF HS(I)=33 THEN GOSUB 4460
530 IF HS(I)=34 THEN GOSUB 4830
540 IF HS(I)=3 THEN GOSUB 5210
550 IF HS(I)=4 THEN GOSUB 5250
560 IF HS(I)=5 THEN GOSUB 5300
570 IF HS(I)=6 THEN GOSUB 5370
580 IF HS(I)=7 THEN GOSUB 5450
590 IF HS(I)=8 THEN GOSUB 5540
600 PRINT:NEXT I
610 PRINT
620 PRINT TAB(10);"TRYCK PÅ RETURN";:INPUT Q$
630 IF L=6 THEN F=9
640 NEXT F
650 END
660 PRINT TAB(10); H$(I,1);
670 PRINT TAB(50); H$(I,2)
680 PRINT TAB(50); H$(I,3)
690 RETURN
700 PRINT TAB(10); H$(I,1);
710 PRINT TAB(50); H$(I,3)
720 PRINT TAB(10); H$(I,2);
730 PRINT TAB(50); H$(I,4)
740 RETURN
750 ON F GOTO 760,800,840,880,920,960,1000,1040,1080
760 FOR J=3 TO 4
770 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-2)
780 NEXT
790 GOTO 1110
800 FOR J=5 TO 6
810 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-4)
820 NEXT
830 GOTO 1110
840 FOR J=1 TO 2
850 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J+2)
860 NEXT
870 GOTO 1110
880 FOR J=5 TO 6
890 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-4)
900 NEXT
910 GOTO 1110
920 FOR J=1 TO 2
930 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J+2)
940 NEXT
950 GOTO 1110
960 FOR J=3 TO 4
970 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-2)
980 NEXT
990 GOTO 1110
1000 FOR J=1 TO 2
1010 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J+2)
1020 NEXT
1030 GOTO 1110
1040 FOR J=3 TO 4
1050 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-2)
1060 NEXT
1070 GOTO 1110
1080 FOR J=5 TO 6
1090 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-4)
1100 NEXT
1110 RETURN
1120 ON F GOTO 1130,1170,1210,1250,1290,1330,1370,1410,1450
1130 FOR J=5 TO 6
1140 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-4)
1150 NEXT
1160 GOTO 1480
1170 FOR J=1 TO 2
1180 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J+2)
1190 NEXT
1200 GOTO 1480
1210 FOR J=3 TO 4
1220 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-2)
1230 NEXT
1240 GOTO 1480
1250 FOR J=3 TO 4
1260 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-2)
1270 NEXT
1280 GOTO 1480
1290 FOR J=5 TO 6
1300 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-4)
1310 NEXT
1320 GOTO 1480
1330 FOR J=1 TO 2
1340 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J+2)
1350 NEXT
1360 GOTO 1480
1370 FOR J=1 TO 2
1380 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J+2)
1390 NEXT
1400 GOTO 1480
1410 FOR J=3 TO 4
1420 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-2)
1430 NEXT
1440 GOTO 1480
1450 FOR J=5 TO 6
1460 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-4)
1470 NEXT
1480 RETURN
1860 ON F GOTO 1870,1910,1950,1990,2030,2070,2110,2150,2190
1870 FOR J=1 TO 2
1880 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J+2)
1890 NEXT
1900 GOTO 2220
1910 FOR J=3 TO 4
1920 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-2)
1930 NEXT
1940 GOTO 2220
1950 FOR J=5 TO 6
1960 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-4)
1970 NEXT
1980 GOTO 2220
1990 FOR J=1 TO 2
2000 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J+2)
2010 NEXT
2020 GOTO 2220
2030 FOR J=3 TO 4
2040 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-2)
2050 NEXT
2060 GOTO 2220
2070 FOR J=5 TO 6
2080 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-4)
2090 NEXT
2100 GOTO 2220
2110 FOR J=1 TO 2
2120 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J+2)
2130 NEXT
2140 GOTO 2220
2150 FOR J=3 TO 4
2160 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-2)
2170 NEXT
2180 GOTO 2220
2190 FOR J=5 TO 6
2200 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-4)
2210 NEXT
2220 RETURN
2230 REM
2240 ON F GOTO 2250,2290,2330,2370,2410,2450,2490,2530,2570
2250 FOR J=1 TO 2
2260 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J+2)
2270 NEXT
2280 GOTO 2600
2290 FOR J=1 TO 2
2300 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J+2)
2310 NEXT
2320 GOTO 2600
2330 FOR J=1 TO 2
2340 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J+2)
2350 NEXT
2360 GOTO 2600
2370 FOR J=3 TO 4
2380 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-2)
2390 NEXT
2400 GOTO 2600
2410 FOR J=3 TO 4
2420 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-2)
2430 NEXT
2440 GOTO 2600
2450 FOR J=3 TO 4
2460 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-2)
2470 NEXT
2480 GOTO 2600
2490 FOR J=5 TO 6
2500 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-4)
2510 NEXT
2520 GOTO 2600
2530 FOR J=5 TO 6
2540 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-4)
2550 NEXT
2560 GOTO 2600
2570 FOR J=5 TO 6
2580 PRINT TAB(10); H$(I,J);:PRINT TAB(50);H$(I,J-4)
2590 NEXT
2600 RETURN
```

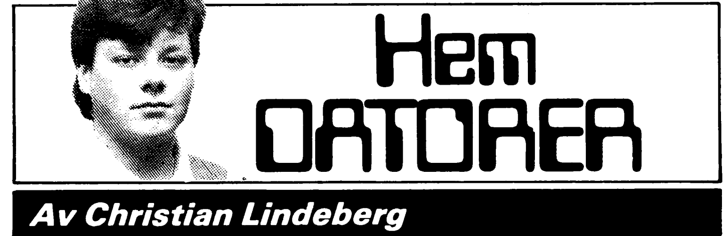
Reducerat V 65-system (2)

Den här veckan fortsätter jag med programmet för att göra reducerade system till V 65. Hela programmet är så långt att det måste delas upp på tre veckor. Systemet går ut på att man delar upp ett stort system på nio st V 65-kuponger och låter datorn reducera till lämpligt antal rader.

I systemet finns alla varianter från tre till nio hästar, vidare finns möjligheten att använda matematiska garderingar från en till åtta hästar.

Förra veckan fick du början på programmet med två säkra hästar och sex hästar i övriga lopp. Denna vecka kommer du att få en komplettering med att rotera nio hästar i fyra lopp med två säkra. Kom ihåg att använda samma radnummer, som jag har gjort för annars stämmer inte programmen när du sätter ihop dem.

När du skriver in det nya programmet så laddar du först in det program du skrev förra veckan, sedan börjar du att knappa in denna vecka program. När du är klar så sparar du programmet. Nästa vecka kommer jag med slutet på programmet och då får du de matematiska graderingarna och rotering av tre hästar.



Av Christian Lindeberg

```
1490 ON F GOTO 1500,1540,1580,1620,1660,1700,1740,1780,1820
1500 FOR J=1 TO 3
1510 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J+3)
1520 NEXT
1530 GOTO 1850
1540 FOR J=1 TO 3
1550 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J+3)
1560 NEXT
1570 GOTO 1850
1580 FOR J=1 TO 3
1590 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J+3)
1600 NEXT
1610 GOTO 1850
1620 FOR J=4 TO 6
1630 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-3)
1640 NEXT
1650 GOTO 1850
1660 FOR J=4 TO 6
1670 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-3)
1680 NEXT
1690 GOTO 1850
1700 FOR J=4 TO 6
1710 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-3)
1720 NEXT
1730 GOTO 1850
1740 FOR J=7 TO 9
1750 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-6)
1760 NEXT
1770 GOTO 1850
1780 FOR J=7 TO 9
1790 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-6)
1800 NEXT
1810 GOTO 1850
1820 FOR J=7 TO 9
1830 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-6)
1840 NEXT
1850 N=0:PRINT:RETURN
2610 ON F GOTO 2620,2660,2700,2740,2780,2820,2860,2900,2940
2620 FOR J=1 TO 3
2630 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J+3)
2640 NEXT
2650 GOTO 2970
2660 FOR J=4 TO 6
2670 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-3)
2680 NEXT
2690 GOTO 2970
2700 FOR J=7 TO 9
2710 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-6)
2720 NEXT
2730 GOTO 2970
2740 FOR J=1 TO 3
2750 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J+3)
2760 NEXT
2770 GOTO 2970
2780 FOR J=4 TO 6
2790 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-3)
2800 NEXT
2810 GOTO 2970
2820 FOR J=7 TO 9
2830 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-6)
2840 NEXT
2850 GOTO 2970
2860 FOR J=1 TO 3
2870 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J+3)
2880 NEXT
2890 GOTO 2970
2900 FOR J=4 TO 6
2910 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-3)
2920 NEXT
2930 GOTO 2970
2940 FOR J=7 TO 9
2950 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-6)
2960 NEXT
2970 N=0:PRINT:RETURN
2980 ON F GOTO 2990,3030,3070,3110,3150,3190,3230,3270,3310
2990 FOR J=4 TO 6
3000 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-3)
3010 NEXT
3020 GOTO 3340
3030 FOR J=7 TO 9
3040 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-6)
3050 NEXT
3060 GOTO 3340
3070 FOR J=1 TO 3
3080 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J+3)
3090 NEXT
3100 GOTO 3340
3110 FOR J=7 TO 9
3120 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-6)
3130 NEXT
3140 GOTO 3340
3150 FOR J=1 TO 3
3160 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J+3)
3170 NEXT
3180 GOTO 3340
3190 FOR J=4 TO 6
3200 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-3)
3210 NEXT
3220 GOTO 3340
3230 FOR J=1 TO 3
3240 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J+3)
3250 NEXT
3260 GOTO 3340
3270 FOR J=4 TO 6
3280 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-3)
3290 NEXT
3300 GOTO 3340
3310 FOR J=7 TO 9
3320 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-6)
3330 NEXT
3340 N=0:PRINT:RETURN
3350 ON F GOTO 3360,3400,3440,3480,3520,3560,3600,3640,3680
3360 FOR J=7 TO 9
3370 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-6)
3380 NEXT
3390 GOTO 3710
3400 FOR J=1 TO 3
3410 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J+3)
3420 NEXT
3430 GOTO 3710
3440 FOR J=4 TO 6
3450 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-3)
3460 NEXT
3470 GOTO 3710
3480 FOR J=4 TO 6
3490 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-3)
3500 NEXT
3510 GOTO 3710
3520 FOR J=7 TO 9
3530 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-6)
3540 NEXT
3550 GOTO 3710
3560 FOR J=1 TO 3
3570 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J+3)
3580 NEXT
3590 GOTO 3710
3600 FOR J=1 TO 3
3610 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J+3)
3620 NEXT
3630 GOTO 3710
3640 FOR J=4 TO 6
3650 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-3)
3660 NEXT
3670 GOTO 3710
3680 FOR J=7 TO 9
3690 PRINT TAB(10); H$(I,J);:N=N+1:IF N<=2 THEN PRINT TAB(50);H$(I,J-6)
3700 NEXT
3710 N=0:PRINT:RETURN
```

Banta programmen med matriser

Jag skall försöka krypa upp ur brevhögarna. När jag startade med V 65-systemen trodde jag inte att intresset var så stort.

Jag skall visa en lösning, som jag fått ifrån en läsare i Ystad. Han har bantat mitt program ner till 2746 bytes genom att använda matriser och DATA-satser.



Av Christian Lindeberg

I raderna 40 till 70 ligger matrisen, som ger ordningen på de olika kupongerna. I raden 90 ligger matrisen som tar ut reserver beroende på vilket system man har begärt.

De olika roterande systemen ligger i raderna 790—900.

Raderna 790—820 roterar tre hästar.

Raderna 830—860 roterar sex hästar.

Raderna 870—900 roterar nio hästar.

De matematiska systemen ligger i raderna 910 till 970.

Jag har numrerat om programmet och ändrat på en del små saker.

Mitt första program var uppe i cirka 15 tusen bytes och läsarens är nere i 2746 bytes. För er som programmerar mycket kan det vara en bra lärdom att studera våra program. Det händer ofta att man genom att arbeta i matriser kan minska ett program till mindre än hälften. Jag kommer lite längre fram att visa exempel på hur man arbetar med matriser både tvådimensionellt, men även med flerdimensionella matriser. Nu till en del frågor, som har kommit den här veckan.

En läsare i Höganäs undrar om han kan skriva av programmet direkt till en Commodore PC-20 med MS-dos. Svaret är ja. Mina program är skrivna på Commodore PC 20.

En annan läsare i Landskrona har en AMSTRAD PCW 8256, som arbetar under CP/M 3.0. Datorn är inte kompatibel med en PC-dator. När det gäller frågan om "copyright", så får man inte publicera ett program från en annan tidning utan att ha tillstånd.

Byta program

Jag vill gärna utveckla det här med att byta program i basic eller andra språk mellan datorer som inte är kompatibla. Jag har många gånger bytt program mellan olika icke kompatibla datorer. Vad som behövs är att man har ett modem och att man kan spara sin programfil i ASCII-format. I en hel del basicdialekter kan man spara sitt program genom att efter sista citationstecknet skriva, A. Så här, SAVE "PROGRAMNAMN", A och nu lagrar datorn det som ASCII-tecken. Nu kan man skicka via telefon till en annan dator och köra den i sitt basicspråk. Är man lite teknisk kan man enkelt göra en kabel mellan datorernas seriella port och köra över det ganska snabbt. Det går bra att köra ifrån en VIC-64 till en PC, men inte tvärtom för 64:an kan inte läsa ASCII-format. När man för över till en ASCII-fil på 64:an så gör man så här.

OPEN 8,8,8, "PROGRAMNAMN,S,W":CMD 8:LIST Tryck på RETURN PRINT#8

CLOSE 8 Tryck på RETURN

Nu sparar datorn ditt program, som en ASCII-fil.

Till slut skall jag ta en fråga från en läsare i Eslöv. Hans problem är att han har en ABC-80 och vill skriva in V 65-programmet. Pröva det tidigare beskrivna programmet från läsaren i Ystad så har du inga problem med minnesutrymme. En sak du måste man tänka på ABC-80 har en lite annorlunda dialekt. Det är därför man får ERR 11 när man försöker skriva in i mitt program. På din datorn går det inte att skriva INPUT-satser med text så rad 430 ändras till:

430 PRINT:PRINT TAB(10); "ÄR DETTA OK (J/N)";:INPUT Q$

På alla ställe det står STR$ ändras till NUM$.

```
10 CLS
20 DIM H$(6,15),A1(4,9),A2(4,9),A3(4,9),P(6,30)
30 FOR K=1 TO 4:FOR L=1 TO 9:READ A(K,L):NEXT L:NEXT K
40 DATA 1,1,1,2,2,2,3,3,3
50 DATA 1,2,3,1,2,3,1,2,3
60 DATA 2,3,1,3,1,2,1,2,3
70 DATA 3,1,2,2,3,1,1,2,3
80 FOR K=1 TO 3:FOR L=1 TO 3:READ A1(K,L),A2(K,L),A3(K,L):NEXT L:NEXT K
90 DATA 1,1,1,2,2,3,3,2,3,2,3,4,1,4,6,3,-2,-3,3,5,7,1,6,9,2,-4,-6
100 PRINT:PRINT:PRINT TAB(10);"INMATNING AV HÄSTAR TILL V65"
110 PRINT:PRINT TAB(10);"MATA IN HÄSTENS NUMMER OCH NAMN"
120 PRINT:PRINT TAB(10);"I DE SEX OLIKA LOPPEN"
130 PRINT:PRINT TAB(10);"OCH I DIN RANKING ORDNING"
140 PRINT:PRINT TAB(10);"I VARJE LOPP SLAR DU IN
150 PRINT:PRINT TAB(10);"MAX 15 HÄSTAR ELLER MINDRE BEROENDE"
160 PRINT:PRINT TAB(10);"PA VILKET SYSTEM DU VILL HA"
170 FOR I=1 TO 6
180 PRINT TAB(10)"LOPP NUMMER";I
190 PRINT TAB(10);"HUR MANGA HÄSTAR";:INPUT N
200 FOR J=1 TO N
210 PRINT TAB(10);"HÄST NUMMER";J;:INPUT H$(I,J)
220 NEXT J
230 NEXT I
240 CLS:PRINT TAB(10);"REUCERADE SYSTEM MED FöLJANDE KOMBINATIONER"
250 PRINT:PRINT TAB(10);"HÄSTARNA KAN SPIKAS GENOM ATT ANGE 1"
260 PRINT:PRINT TAB(10);"3 ST HÄSTAR SOM ROTERAR EN OCH EN HETER 31,32,33 OCH 34
"
270 PRINT:PRINT TAB(10);"6 ST HÄSTAR SOM ROTERAR TVÅ OCH TVÅ HETER 61,62,63 OCH
64"
280 PRINT:PRINT TAB(10);"9 ST HÄSTAR SOM ROTERAR TRE OCH TRE HETER 91,92,93 OCH
94"
290 PRINT:PRINT TAB(10);"MATEMATISKA GARDERINGAR SKRIV 2, 3, 4, 5, 6, 7 ELLER 8"
300 PRINT:PRINT TAB(10);"TÄNK PA ATT OM DU ANVÄNDER ROTERANDE SYSTEM"
310 PRINT:PRINT TAB(10);"MASTE ENTALSSIFFRORNA VARA I FöLJD 31,62,93,64"
320 D=1:G=1:L=0
330 FOR I=1 TO 6
340 PRINT:PRINT TAB(10);"VILKEN KOMBINATION I LOPP";I;:INPUT HS(I)
350 IF HS(I)<10 THEN L=L+1
360 IF LEN(STR$(HS(I)))>2 THEN D=D*VAL(MID$(STR$(HS(I)),2,1))
370 IF LEN(STR$(HS(I)))>2 THEN G=G*(VAL(MID$(STR$(HS(I)),2,1)))/3
380 IF LEN(STR$(HS(I)))=2 THEN D=D*HS(I)
390 IF LEN(STR$(HS(I)))=2 THEN G=G*HS(I)
400 NEXT I
410 PRINT:PRINT TAB(10);"SYSTEMET ÄR PÅ";D" RADER OCH REDUCERAS TILL";G*9"RADER"
420 PRINT:PRINT TAB(10);"OCH KOSTAR"G*9*.7"KRONOR"
430 PRINT:PRINT TAB(10);:INPUT "ÄR DETTA OK (J/N)",Q$
440 IF Q$="J" OR Q$="j" THEN 460
450 GOTO 240
460 FOR F=1 TO 9
470 CLS:PRINT:PRINT"LAPP NR.";F
480 PRINT
490 FOR I=1 TO 6
500 PRINT I;
510 IF HS(I)=1 THEN GOSUB 620
520 IF HS(I)=2 THEN GOSUB 660
530 IF LEN(STR$(HS(I)))>2 THEN GOTO 710
540 IF HS(I)>=3 AND HS(I)<=8 THEN GOSUB 910
550 PRINT:NEXT I
560 PRINT
570 INPUT Q$
580 IF L=6 THEN F=9
590 Z=0
600 NEXT F
610 END
620 PRINT TAB(10); H$(I,1);
630 PRINT TAB(50); H$(I,2)
640 PRINT TAB(50); H$(I,3)
650 RETURN
660 PRINT TAB(10); H$(I,1);
670 PRINT TAB(50); H$(I,3)
680 PRINT TAB(10); H$(I,2);
690 PRINT TAB(50); H$(I,4)
700 RETURN
710 Z=Z+1
720 Y=VAL(LEFT$(STR$(HS(I)),2))
730 IF Y=3 THEN GOSUB 790 ELSE 750
740 GOTO 550
750 IF Y=6 THEN GOSUB 830 ELSE 770
760 GOTO 550
770 GOSUB 870
780 GOTO 550
790 PRINT TAB(10);H$(I,A1(A(Z,F),1));
800 PRINT TAB(50);"R";H$(I,A1(A(Z,F),2))
810 PRINT TAB(50);"R";H$(I,A1(A(Z,F),3))
820 RETURN
830 FOR J=A2(A(Z,F),1) TO A2(A(Z,F),2)
840 PRINT TAB(10);H$(I,J);TAB(50);"R";H$(I,J+A2(A(Z,F),3))
850 NEXT J
860 RETURN
870 N=0:FOR J=A3(A(Z,F),1) TO A3(A(Z,F),2)
880 PRINT TAB(10);H$(I,J);:N=N+1: IF N<=2 THEN PRINT TAB(50);"R";H$
    (I,J+A3(A(Z,F ),3));
890 PRINT:NEXT J
900 RETURN
910 FOR J=1 TO HS(I)
920 PRINT TAB(10);H$(I,J);
930 IF J=HS(I)-1 THEN PRINT TAB(50);"R";H$(I,HS(I)+1);
940 IF J=HS(I) THEN PRINT TAB(50);"R";H$(I,HS(I)+2);
950 PRINT
960 NEXT J
970 RETURN
```

Datorn hjälper dig med matten

Den här veckan skall jag visa ett program om matematik. Jag fick ett brev från Tobias W i Kävlinge. Tobias håller på att göra en del nyttoprogram till sin VIC 64:a och skulle gärna vilja komplettera med ett matteprogram.



Av Christian Lindeberg

Jag är lite undrande över ditt sätt att beskriva vad du vill ha men jag har förutsatt att du strök under lite för många tal i ditt brev. I annat fall så använder du datorn bara som en bit papper.

Komprimerat

Jag har gjort ett program, som tillåter träning av huvudräkning till vilken övre gräns som helst. Själva räkneprogrammet är komprimerat. Man kan göra det större genom att skriva om alla rutinerna fyra gånger och ändra räknesätten. Jag skall förklara programmet, som jag gjort.

På rad 10 och 20 har jag använt mig av funktionen DEF FN, som de flesta är lite rädda för att använda. Jag skall förklara hur den fungerar. I rad 10 och 20 har jag definierat funktionen DEF FNA(X-)=(INT(RND*X)+1). Detta betyder att när jag sedan i programmet skriver TALI= FNA(X), så letar datorn reda på vad X har för värde och sätter in detta värde i den fördefinierade funktionen i rad 10. Själva funktionen INT(RND*X)+1 ger ett värde mellan 1 och X. Tänk på att på VIC-datorerna skriver du inte RND utan RND(0).

Högsta gräns

Rad 30 tömmer skärmen. Skriv PRINT CHR$(147) på VIC-arna. Har du en annan dator så kontrollera med din instruktionsbok.

Rad 40 till 110 förklarande text till användaren av programmet. Rad 120 hoppar till rad 350 när man vill avsluta och ha reda på sitt resultat.

Raderna 130 till 170 förklarar för användaren hur man skall sätta sina gränser. Talet du slår in i rad 170 blir den högsta gränsen för talen till addition, subtraktion och multiplikation. När det gäller division så måste jag svänga lite på variablerna för att jag skall få heltal. Detta gör att när jag skall köra division så måste jag tänka på att det tal jag slår in i rad 170 blir kvadratiskt. Om du slår in 10 så blir dina högsta divisioner 100 delat med något. Om du vill ha olika gränser på de två talen så ändra rad 170 till:
170 INPUT X,Y

Kommatecken

Men nu måste du slå in två tal med ett kommatecken emellan.

Rad 180 tömmer skärmen.

Rad 190 ger informationen att när du vill sluta så slå in —99.

Rad 200 räknar hur många tal du har gjort.

Rad 210 bestämmer de två olika talen.

Rad 220 till 250 IF-satser, som datorn hoppar till beroende på vad du valde i raderna 60 till 90. Rad 250 ser lite annorlunda ut — detta beror på att jag vill ha heltal i mina divisioner. Jag tar först reda på vad multiplikationen med TAL1 och TAL2 blir och lägger svaret i variabeln W. Sedan i rad 290 delar jag W med TAL1 och svaret blir TAL2. Det är därför jag har skrivit i slutet på rad 250 SVAR=TAL2.

Tre försök

Rad 260 till 290 skriver ut de olika matematiska frågeställningarna beroende av vad jag har valt i raderna 60 till 90.

På rad 300 svarar du datorn vad du tror att svaret blir.

Rad 310 kontrollerar om du vill gå tillbaka till huvudmenyn.

Rad 320 kontrollerar om ditt svar är rätt och låter dig fortsätta med en ny uppgift.

På rad 330 här hamnar man om man har gjort fel. Datorn tillåter tre försök innan den talar om det rätta svaret och låter dig gå vidare.

Rad 340 ett hopp till rad 220 för ett nytt problem.

Rad 350 tömmer skärmen.

Rad 360 till 370 talar om hur duktig du har varit. Här kan du själv komplettera med rader som ger dig olika mycket beröm beroende på hur duktig du har varit.

```
10 DEF FN A(X)=(INT(RND*X)+1)
20 DEF FN B(Y)=(INT(RND*Y)+1)
30 CLS
40 PRINT:PRINT TAB(10)"HUVUDRÄKNING"
50 PRINT:PRINT
60 PRINT TAB(10)"1. ADDITION"
70 PRINT:PRINT TAB(10)"2. SUBTRAKTION"
80 PRINT:PRINT TAB(10)"3. MULTIPLIKATION"
90 PRINT:PRINT TAB(10)"4. DIVISION"
100 PRINT:PRINT TAB(10)"5. AVSLUTA"
110 PRINT:PRINT TAB(10)"VAD öNSKAS";:INPUT Q$
120 IF VAL(Q$)=5 THEN 350
130 PRINT:PRINT TAB(10)"BESTÄM STORLEKEN PÅ TALEN"
140 PRINT:PRINT TAB(10)"SKRIV TALENS HöGSTA GRÄNS"
150 PRINT:PRINT TAB(10)"VID DIVISION BLIR GRÄNSEN TALET"
160 PRINT:PRINT TAB(10)"MULTIPLICERAT MED SIG SJÄLV ";
170 INPUT X:Y=X
180 CLS
190 PRINT:PRINT TAB(10)"NÄR DU VILL SLUTA SÅ SLÅ IN -99"
200 N=N+1
210 TAL1=FNA(X):TAL2=FNB(Y)
220 IF VAL(Q$)=1 THEN SVAR=TAL1+TAL2
230 IF VAL(Q$)=2 THEN SVAR=TAL1-TAL2
240 IF VAL(Q$)=3 THEN SVAR=TAL1*TAL2
250 IF VAL(Q$)=4 THEN W=TAL1*TAL2:SVAR=TAL2
260 IF VAL(Q$)=1 THEN PRINT:PRINT TAB(10)"VAD BLIR"TAL1"+"TAL2;
270 IF VAL(Q$)=2 THEN PRINT:PRINT TAB(10)"VAD BLIR"TAL1"-"TAL2;
280 IF VAL(Q$)=3 THEN PRINT:PRINT TAB(10)"VAD BLIR"TAL1"*"TAL2;
290 IF VAL(Q$)=4 THEN PRINT:PRINT TAB(10)"VAD BLIR"W"/"TAL1;
300 INPUT TEST
310 IF TEST=-99 THEN 30
320 IF TEST=SVAR THEN PRINT:PRINT TAB(10)"RÄTT";:INPUT W$:GOTO
                                                            180
330 PRINT:PRINT TAB(10)"FEL FöRSöK IGEN";:FEL=FEL+1:IF FEL=3
                                          THEN PRINT:PRINT TA
B(10)"RÄTT SVAR ÄR";SVAR;:FEL=0:F=F+1:INPUT W$:GOTO 180
340 GOTO 220
350 CLS
360 PRINT:PRINT:PRINT TAB(10) "DU HADE"N-F"RÄTT OCH"F"FEL"
370 PRINT:PRINT TAB(10)"TACK FöR DENNA GÅNG"
```

Lekande lätt med matriser



Hem DATORER

Av Christian Lindeberg

Den här veckan skall jag tala om matriser. Ni kommer kanske ihåg programmet om reducerade V 65-system, där programmet gick att banta radikalt genom att använda matriser i stället för att upprepa liknande programdelar ett flertal gånger.

Vad är en matris? Matriser är något, som man använder i matematiken när man skall lösa ekvationssystem med många obekanta. Vi skall inte lösa matematiska problem utan se matrisen från en annan vinkel. Jag tänkte ta exemplet med ett register där man har namn, adress, postnummer, postadress och telefonnummer. När man gör ett register kan man använda variblerna E$(I), A$(I), P$(I), PA$(I) och T$(I). För att se tillhörigheten skriver jag så här:

Namn	Adress	Postadress	Postnummer	Telefon
E$(I)	A$(I)	P$(I)	PA$(I)	T$(I)

Det vi har här är en matris med kolumnerna NAMN, ADRESS, POSTADRESS, POSTNUMMER och TELEFON. Raderna blir sedan varje namn med uppgifter om personen. Antalet rader blir lika med antalet namn. Nu har vi utgångspunkten för en matris, nämligen en del kolumner och en del rader. Nu skall vi se på problemet ur en annan vinkel.

Namn	Adress	Postadress	Postnummer	Telefon
E$(I)	A$(I)	P$(I)	PA$(I)	T$(I)
Kolumn 1 E$(I,1)	Kolumn 2 E$(I,2)	Kolumn 3 E$(I,3)	Kolumn 4 E$(I,4)	Kolumn 5 E$(I,5)

Jag har bytt ut alla variablerna mot en variabel, som heter E$(I,J). Där I är antalet namn och J är de olika kolumnerna 1 till 5.

De av er som har ett bra minne kommer kanske ihåg ett program, som jag har haft om register, där man kunde söka på namn, adress, postnummer och telefonnummer. Programmet var på 70 rader och hade många rutiner som upprepades beroende på att jag hade använt variablerna E$(I), A$(I), P$(I), PA$(I) och T$(I).

Jag har gjort om programmet och använder bara en variabel, nämligen E$(I,J). Programmet minskar nu ifrån 70 rader till 39 rader. I förra programmet hade jag 4 sökrutiner men nu klarar jag det med en sökrutin. Nu över till programmet och förklaringar.

Rad 10. Tömmer skärmen.

Rad 20. Variabeln N är antalet namn, som registret innehåller.

Rad 30. Om man har många uppgifter att läsa in så att datorn tar lång tid på är det alltid trevligt att få reda på att man skall vänta ett tag.

Rad 40. Dimensionering av matrisvariabeln E$(N,5). N får värdet i rad 20. Fem är det antal kolumner, som jag använder (se ovan).

Rad 50. Hopp till en subrutin, som läser in alla uppgifter, som finns i DATA-satserna i raderna 380 och vidare.

Rad 60 till 130. Här ligger registermenyn, där du kan tala om vad du vill söka på.

Rad 140. Här anger du med en siffra vad du vill söka på. Kom ihåg, ni som har ABC-80 och liknande datorer ni kan inte skriva INPUT med text utan måste ändra rad 140 så här:

140 PRINT:PRINT:PRINT "VAD VILL DU SÖKA PÅ (1, 2, 3, 4, 5 ELLER 6);:INPUT A.

Rad 150. Känner av om variabeln A är lika med 6 för då hoppar datorn ur programmet.

Rad 160 till 210. Sökrutinen för hela programmet, här anger du va namnet, adressen eller vad du vill söka efter. Vill du söka efter mitt telefonnummer så skriver du bara 45 54 51 och sedan söker datorn upp alla uppgifter och skriver ut dem i raderna 230 till 310. Lägg märke till rad 200 där jag har använt det värde du matade in i variabeln A i rad 140, eftersom detta värde också stämmer överens med den kolumn, som innehåller motsvaande information.

Rad 220. Hoppar tillbaka till menyn när du är klar.

Rad 230 till 310. Skriver ut alla uppgifter om den du sökte på.

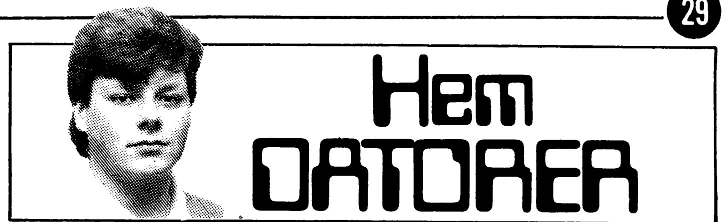
Rad 320 till 370. En rutin som läser in alla uppgifter som ligger i DATA-satserna i raderna 380 och framåt.

Rad 380 och framåt. Här bygger du på med namn och uppgifter på dina kompisar.

GLÖM INTE ATT ÖKA VARIABELN N I RAD 20!

```
10 CLS
20 N=2:REM ANTAL NAMN
30 PRINT:PRINT:PRINT:PRINT"                     VAR  GOD  VÄNTA"
40 DIM E$(N,5)
50 GOSUB 320
60 CLS
70 PRINT"   REGISTERMENY"
80 PRINT"1. SöKNING PÅ NAMN"
90 PRINT"2. SöKNING PÅ ADRESS"
100 PRINT"3. SöKNING PÅ POSTNUMMER"
110 PRINT"4. SöKNING PÅ POSTADRESS"
120 PRINT"5. SöKNING PÅ TELEFONNUMMER"
130 PRINT"6. AVSLUTA"
140 PRINT:PRINT:INPUT"VAD VILL DU SöKA PÅ (1,2,3,4,5 ELLER 6)";A
150 IF A=6 THEN END
160 CLS
170 PRINT"SKRIV NAMNET ELLER NUMRET DU SöKER PÅ ";
180 INPUT B$
190 FOR I=1 TO N
200 IF B$=E$(I,A) THEN GOSUB 230
210 NEXT I
220 GOTO 60
230 CLS
240 PRINT:PRINT:PRINT"   NAMN        : ";E$(I,1)
250 PRINT:PRINT"   ADRESS      : ";E$(I,2)
260 PRINT:PRINT"   POSTNUMMER  : ";E$(I,3)
270 PRINT:PRINT"   POSTADRESS  : ";E$(I,4)
280 PRINT:PRINT"   TELEFON     : ";E$(I,5)
290 PRINT:PRINT:PRINT"   TRYCK PÅ RETURN";
300 INPUT QQ$
310 RETURN
320 FOR I=1 TO N
330 FOR J=1 TO 5
340 READ E$(I,J)
350 NEXT J
360 NEXT I
370 RETURN
380 DATA LINDEBERG CHRISTIAN,SANDJORDSGRÄND 10,236 00,HöLLVIKEN,45 53 51
390 DATA DATASSON FLOPPY,DISKETTVÄGEN 2,999 99, CHIPHöGEN, 66 99 33
```

Så sorterar man använda matriser



Av Christian Lindeberg

Förra veckan visade jag hur man kunde förenkla ett registerprogram genom att använda en matrismodell. Ett följdproblem av att använda matriser är att sortera. Hur sorterar man när man har två index i en variabel? Denna fråga skall vi lösa i dag.

Problem nummer ett är att man måste ha klart för sig vad de olika variablerna står för. En kort repetition av förra veckan.

Namn E$(I)	Adress A$(I)	Postadress P$(I)	Postnummer PA$(I)	Telefon T$(I)
Kolumn 1	Kolumn 2	Kolumn 3	Kolumn 4	Kolumn 5
E$(I,1)	E$(I,2)	E$(I,3)	E$(I,4)	E$(I,5)

Här har vi de fem olika variablerna från E$(I,1) till E$(I,5). I första indexet är person nummer 1 till hur många jag nu har i mitt register. I programmet har jag använt variabeln N. Det andra indexet är de olika uppgifterna om varje person. 1 är namn, 2 är adress, 3 är postadress, 4 är postnummer och 5 är telefonnummer. Om jag nu skall sortera namn så måste det andra indexet hela tiden vara 1. Alltså jag skall sortera variabeln E$(I,1). Om jag skall sortera telefonnummer blir det variabeln E$(I,5). I programmet jag har skrivit den här gången så har jag använt mig av en s k Quick-sort rutin för att det skall gå snabbt. Låt oss gå igenom programmet.

Ny punkt

Den första delen är nästan samma som förra gången. Jag har lagt till en punkt till i menyn där man kan sortera och få utskrift på skärm. Vi skall bara titta på sorteringsrutinen. När man väljer alternativ 6 i huvudmenyn så hamnar man i rad 670, som visar upp en sorteringsmeny. Här kan du välja vad du vill sortera på. Ditt val läggs i variabeln B. Denna variabel innehåller nu en siffra, som motsvarar det du vill sortera på. Om du har valt "5 Telefon" i menyn, så ger siffran 5 också kolumm 5 se ovan. Nu hoppar datorn till rad 420 där det ligger en Quicksort-rutin. Quicksortrutinen är ganska komplicerad, så jag hoppar över förklaringen, men jag tar upp ett par viktiga punkter. I rad 450 där är E$(L,B), det sökbegrepp du angett i rad 750. Indexet L är antalet personer och B är den uppgift du angett i sorteringsmenyn.

Byter plats

I rad 510 till 530 byter personerna och deras uppgifter plats om de inte befinner sig i rätt ordning. Sorteringsprogrammet sorterar från det lägsta till det största. Om du vill ändra så att datorn sorterar tvärtom så ändra tecknen i raderna 460 och 480 så att större än och mindre än tecknen blir tvärtom. När datorn sorterat färdigt så hoppar den till rad 780 och frågar vilken mer uppgift du vill ha med när du får utskriften på skärmen. Att det inte blir mer komplicerat beror på att vi använder oss av matrisvariabler. Det viktigaste är att hålla tungan rätt i mun, när man hanterar variablerna och inte blandar ihop dem. Ett bra tips är att göra upp en matris på ett papper och skriva in sina variabler och vad de handlar om. Lycka till!

Skriv gärna

till Christian Lindeberg om allt som har med hemdatorer att göra.

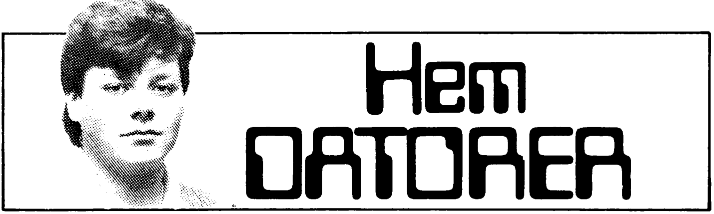
Adressen är:

Hemdatorer
Arbetet
Box 125
201 21 Malmö

```
10 CLS
20 N=2:REM ANTAL NAMN
30 PRINT:PRINT:PRINT:PRINT"
                      VAR GOD  VÄNTA"
40 DIM E$(N,5)
50 GOSUB 360
60 GOSUB 420
70 CLS
80 PRINT"   REGISTERMENY"
90 PRINT"1. SÖKNING PÅ NAMN"
100 PRINT"2. SÖKNING PÅ ADRESS"
110 PRINT"3. SÖKNING PÅ POSTNUMMER"
120 PRINT"4. SÖKNING PÅ POSTADRESS"
130 PRINT"5. SÖKNING PÅ TELEFONNUMMER"
140 PRINT"6. SORTERING OCH UTSKRIFT PÅ SKÄRM"
150 PRINT"7. AVSLUTA"
160 PRINT:PRINT:INPUT"VAD VILL DU SÖKA PÅ
                  (1,2,3,4,5 ELLER 6)";A
170 IF A=6 THEN 670
180 IF A=7 THEN END
190 IF A<1 OR >5 THEN 70
200 CLS
210 PRINT"SKRIV NAMNET ELLER NUMRET DU SÖKER
220 INPUT B$
230 FOR I=1 TO N
240 IF B$=E$(I,A) THEN GOSUB 270
250 NEXT I
260 GOTO 70
270 CLS
280 PRINT:PRINT:PRINT"  NAMN        : ";E$(I,1)
290 PRINT:PRINT"   ADRESS      : ";E$(I,2)
300 PRINT:PRINT"   POSTNUMMER  : ";E$(I,3)
310 PRINT:PRINT"   POSTADRESS  : ";E$(I,4)
320 PRINT:PRINT"   TELEFON     : ";E$(I,5)
330 PRINT:PRINT:PRINT"   TRYCK PÅ RETURN";
340 INPUT QQ$
350 RETURN
360 FOR I=1 TO N
370 FOR J=1 TO 5
380 READ E$(I,J)
390 NEXT J
400 NEXT I
410 RETURN
420 F=1:L(1)=1:R(1)=N
430 L=L(F):R=R(F):F=F-1
440 I=L:J=R
450 C$=E$(L,B)
460 IF E$(I,B)>=C$ THEN 480
470 I=I+1:GOTO 460
480 IF E$(J,B)<=C$ THEN 500
490 J=J-1:GOTO 480
500 IF I>J THEN 550
510 FOR K=1 TO 5
520 X$=E$(I,K):E$(I,K)=E$(J,K):E$(J,K)=X$
530 NEXT K
540 I=I+1:J=J-1
550 IF I<=J THEN 460
560 IF J-L>=R-I THEN 610
570 IF I>=R THEN 590
580 F=F+1:L(F)=I:R(F)=R
590 R=J
600 GOTO 640
610 IF L>=J THEN 630
620 F=F+1:L(F)=L:R(F)=J
630 L=I
640 IF L<R THEN 440
650 IF F>0 THEN 430
660 RETURN
670 CLS
680 PRINT"SORTERING "
690 PRINT"VAD VILL DU SORTERA PÅ"
700 PRINT"1. NAMN"
710 PRINT"2. ADRESS"
720 PRINT"3. POSTNUMMER"
730 PRINT"4. POSTADRESS"
740 PRINT"5. TELEFONNUMMER"
750 PRINT:INPUT B
760 IF B<1 OR B>5 THEN 670
770 GOSUB 420
780 CLS
790 PRINT"UTSKRIFT PÅ SKÄRM"
800 PRINT"ANGE YTTERLIGARE EN POST ATT HA
                        MED I UTSKRIFTEN"
810 PRINT"1. NAMN"
820 PRINT"2. ADRESS"
830 PRINT"3. POSTNUMMER"
840 PRINT"4. POSTADRESS"
850 PRINT"5. TELEFONNUMMER"
860 INPUT C
870 IF C<1 OR C>5 THEN 780
880 CLS:FOR I=1 TO N
890 PRINT E$(I,B),E$(I,C)
900 IF I/15=INT(I/15) THEN INPUT Q$
910 NEXT I
920 PRINT:INPUT Q$
930 GOTO 70
940 DATA LINDEBERG CHRISTIAN,SANDJORDSGRÄND 10,
               236 00,HÖLLVIKEN,45 53 51
950 DATA DATASSON FLOPPY,DISKETTVÄGEN 2,
               999 99, CHIPHÖGEN, 66 99 33
```

Skriv program utan datorhjälp

Lovet förra veckan tillbringade jag en del tid på tåg. Den tiden använde jag till att skriva ihop några små program. Det är rätt intressant att skriva program utan dator. När jag kom hem skrev jag av programmen och körde dem i datorn.



Av Christian Lindeberg

Programmen är gjorda från uppgifter i boken "Att chansa med räknedosa" av Lennart Råde. Boken innehåller massor av uppgifter så försök att låna boken på biblioteket, för den har mycket att ge. Boken är skriven för programmerbara räknedosor, men du får inga färdiga program. I en av uppgifterna hittade jag ett exempel där man hade räknat ut sannolikheter för 1 par, 2 par, triss och kåk. Då fick jag uppslaget till första programmet. Hur kan datorn kontrollera vilket det är?

Yatzy

Nästa program var ett Yatzyprogram där man skulle kunna slå om de tärningar som man inte ville ha kvar. Jag skall förklara listorna men först ett par brev.

Det har kommit en del brev om registerprogrammet med sortering. Ni som inte får det att fungera har datorer, som måste ha en dimensionering på alla variabler som är indexerade.

En indexerad variabel är till exempel E(I), där I är ett tal. I de flesta datorer kan man använda indexerade variabler mellan 0 till 10 utan att behöva göra någon dimensionering. Jag kan alltså använda variabeln E(I) från E(0) till E(10). Skall jag ha 11 eller flera index så måsta jag dimensionera. Till exempel med 25 st index blir det att i början av programmet skriva en rad DIM E(25).

```
10 RANDOMIZE(TIMER)
20 CLS
30 FOR I=1 TO 10
40 A(I)=0
50 NEXT I
60 FOR I=1 TO 5
70 S=INT(RND*10)+1
80 PRINT S;
90 A(S)=A(S)+1
100 NEXT I
110 PRINT
120 FOR I=1 TO 10
130 IF A(I)>1 THEN B$=B$+STR$(A(I))
140 NEXT I
150 IF LEN(B$)<2 THEN PRINT
    "INGET":END
160 IF LEN(B$)>3 THEN 180
170 PRINT B$+" -TAL":END
180 IF MID$(B$,1,2)=MID$(B$,3,2)
    THEN PRINT "TVA PAR":END
190 PRINT "KAK"

List POKER
```

```
10 CLS
20 RANDOMIZE (TIMER)
30 FOR I=1 TO 5
40 A(I)=INT(RND*6)+1
50 NEXT I
60 FOR I=1 TO 5
70 PRINT TAB(I*5);I;
80 NEXT I
90 PRINT:PRINT
100 FOR I=1 TO 5
110 PRINT TAB(I*5);A(I);
120 NEXT I
130 PRINT:PRINT:PRINT
140 PRINT"HUR MANGA VILL DU BYTA";
150 INPUT K
160 PRINT:PRINT
170 FOR I=1 TO K
180 PRINT"SLA IN POSITIONEN";
190 INPUT B(K)
200 A(B(K))=INT(RND*6)+1
210 PRINT
220 NEXT I
230 PRINT:PRINT
240 FOR I=1 TO 5
250 PRINT TAB(I*5);I;
260 NEXT I
270 PRINT:PRINT
280 FOR I=1 TO 5
290 PRINT TAB(I*5);A(I);
300 NEXT I

List YATZY
```

Dimensionera

Men det finns tyvärr en del datorer där jag måste dimensionera så fort jag använder en indexerad variabel. Så använder jag bara fram E(1) till E(5), så måste jag i början av programmet skriva DIM E(5). Det betyder att L(F) och R(F) i registerprogrammet måste dimensioneras på en del datorer.

Astronomi

Jag har lånat en bok på biblioteket, som heter "The Astronomical Almanac 1987". Efter att ha tittat igenom den med alla dessa formler och tidsangivelser så är det konstigt att några tider blir rätt. Jag kom underfund med att den tid vi använder är mer eller mindre ett politiskt beslut och den verkliga soltiden är något helt annat. Jag är för tillfället inte kompetent att reda ut begreppen men jag kanske återkommer längre fram. Nu till de två listorna och först tar jag den, som jag kallat POKER.

Rad 10 måste jag ha med för att få olika serier av slumptal. Alla som skriver i GWBASIC eller någon annan PC-basic måste ha rad 10.

Rad 20 till 50 är av säkerhetsskäl — jag nollställer alla variabler, som jag skall använda.

Slumptalet

Raderna 60 till 100 är en loop, som tar fram de siffror som jag skall undersöka. Rad 70 tar fram slumptalet och rad 90 räknar datorn hur många gånger siffran har kommit upp.

Rad 130 lägger de antalet i en sträng om någon A(S) variabel har fått mer än ett tal. Alltså par eller mer.

Raderna 150 och 160 kontrollerar längden på strängen B$. Är den större än 3 så består den av två par eller ett par och en triss, alltså en kåk och då hoppar den till rad 180 som kontrollerar om det är lika, alltså två tvåor, och då skriver ut "TVÅ PAR". I annat fall måste det vara en KÅK, vilket den skriver ut i rad 90. Nu över till YATZY-programmet, som är längre men lite lättare.

Många loopar

Programmet består av en massa loopar där första loopen mellan rad 30 till 50 ger slumptalen mellan 1 och 6. Nästa loop mellan 60 till 80 ger siffrorna för att visa platsen. Raderna 100 till 120 skriver ut vad tärningarna utvisar. I rad 140 kommer sedan frågan om hur många tärningar du vill slå om och med variabeln K får du nu en ny loop i raderna 170 till 220 där du slår om dina tärningar.

Resultatet skrivs sedan ut i raderna 240 till 300.

Så övar du matte på din hemdator



Hem DATORER

Av Christian Lindeberg

Den här veckan kommer jag att ta upp ett problem, som många av mina läsare har frågat om. Det gäller alla som går i skolan och är intresserade av ämnet matematik. Frågan jag har fått är: ”Hur kan jag göra egna övningsexempel på min dator?”

Bra fråga! Problemet är inte så svårt eftersom programmering går ut på att jag vet både frågan och svaret i princip. Låt mig förklara. Exemplet jag skall använda är att räkna ut ränta, procent, tid eller kapital enligt modellen:
kapital★procent/100★tid=ränta
Ett problem är att tiden kan vara dagar eller månader, men det löser vi med ett par subrutiner. Jag skall nu visa hur man löser ett sådant här programmeringsproblem.

Subrutiner

Först börjar jag med subrutiner, som skall räkna ut alla mina faktorer. Jag delar upp problemet i två subrutiner där en skall ta hand om dagar och den andra om månader. Första subrutinen blir då så här:

```
620 REM DAGAR
630 T=INT (RND★360+30)
640 K=INT (RND★24500+500)
650 P=INT (RND★12+8)
660 R+K★P/100★T/360:R=INT
(R★100+0.5)/100
670 RETURN
```

I rad 630 lägger jag till 30 för att lägsta tiden skall vara 30 dagar. I rad 640 blir lägsta beloppet 500. I rad 650 blir lägsta procenten 8%.
I rad 660 så räknar datorn ut vad räntan i kronor skall bli. Sista delen efter kolon ser till att datorn avrundar med två decimaler.

Alla svaren

Den andra subrutinen blir en kopia med två skillnader. Rad 630 med tiden ändras till månader och rad 660 där måste delningstalet för tiden vara 12 i stället för 360, så raderna blir följande:

```
690 T=INT(RND★12+1)
720 R=K★P/100★T/12:R=INT
(R★100+0.5)/100
```

Resten ser likadant ut. När jag har gjort dessa två subrutiner klara så är problemet löst eftersom att jag nu har svaret på alla frågor. Oavsett om jag vill räkna fram kapital, procent, tid eller räntan i kronor, så har datorn alla svaren.

Fyra rutiner

Det som nu behövs är bara en del skrivrutiner för att ställa de riktiga frågorna till dig. Jag har lagt upp fyra stycken rutiner, som är nästan identiska. De ligger med början på raderna 210, 310, 410 och 510. Den första där får du räkna ut kapitalet, den andra där gäller det procenten, den tredje är kanske det vanligaste där du får räkna ut räntan i kronor och till slut har vi den fjärde där du måste räkna ut tiden. Du får här också ett tillfälle att använda dina kunskaper i ekvationer genom att i alternativen 1, 2 och 4 sätta in x för det efterfrågade.

Har du problem med programmet eller har en bättre och mindre lösning på problemet så skriv. Har du själv några tips på liknande problem så hör av dig.

```
10 CLS
20 REM PROGRAM FöR ÖVNINGAR I RÄNTERÄKNING
30 PRINT:PRINT TAB(10);"MENY TID"
40 PRINT:PRINT TAB(10);"VILL DU RÄKNA MED DAGAR ELLER MANADER"
50 PRINT:PRINT TAB(10);"1. DAGAR"
60 PRINT:PRINT TAB(10);"2. MÅNADER"
70 PRINT:PRINT TAB(10);:INPUT A
80 IF A=1 THEN GOSUB 620
90 IF A=2 THEN GOSUB 680
100 IF A<1 OR A>2 THEN 10
110 CLS
120 PRINT:PRINT TAB(10);"MENY EXEMPEL"
130 PRINT:PRINT TAB(10);"VAD VILL DU RÄKNA UT"
140 PRINT:PRINT TAB(10),"1. (K)APITAL"
150 PRINT:PRINT TAB(10),"2. (P)ROCENT"
160 PRINT:PRINT TAB(10),"3. (R)ÄNTA I KRONOR"
170 PRINT:PRINT TAB(10),"4. (T)ID"
180 PRINT:PRINT TAB(10);:INPUT B
190 ON B GOTO 210,310,410,510
200 GOTO 110
210 CLS
220 PRINT:PRINT TAB(10);"HÄR ÄR DITT RÄKNEEXEMPEL"
230 PRINT:PRINT TAB(10);"PROCENT =";P;" %"
240 PRINT:PRINT TAB(10)"RÄNTA PÅ PERIODEN =";R;" KRONOR"
250 PRINT:PRINT TAB(10);"TIDEN =";T;:IF A=1 THEN PRINT " DAGAR"
    ELSE PRINT "MÅNADER"
260 PRINT:PRINT TAB(10);"TRYCK PÅ RETURN NÄR DU VILL VETA SVARET";
270 INPUT Q$
280 PRINT:PRINT TAB(10);"SVARET =";K;" KRONOR"
290 PRINT:PRINT TAB(10)"TRYCK PÅ RETURN";:INPUT Q$
300 GOTO 10
310 CLS
320 PRINT:PRINT TAB(10);"HÄR ÄR DITT RÄKNEEXEMPEL"
330 PRINT:PRINT TAB(10);"KAPITAL =";K;" KRONOR"
340 PRINT:PRINT TAB(10)"RÄNTA PÅ PERIODEN =";R;" KRONOR"
350 PRINT:PRINT TAB(10);"TIDEN =";T;:IF A=1 THEN PRINT " DAGAR"
    ELSE PRINT "MÅNADER"
360 PRINT:PRINT TAB(10);"TRYCK PÅ RETURN NÄR DU VILL VETA SVARET";
370 INPUT Q$
380 PRINT:PRINT TAB(10);"SVARET =";P;" %"
390 PRINT:PRINT TAB(10)"TRYCK PÅ RETURN";:INPUT Q$
400 GOTO 10
410 CLS
420 PRINT:PRINT TAB(10);"HÄR ÄR DITT RÄKNEEXEMPEL"
430 PRINT:PRINT TAB(10);"KAPITAL =";K;" KRONOR"
440 PRINT:PRINT TAB(10)"PROCENTEN =";P;" %"
450 PRINT:PRINT TAB(10):"TIDEN =";T;:IF A=1 THEN PRINT " DAGAR"
    ELSE PRINT MÅNADER"
460 PRINT:PRINT TAB(10);"TRYCK PÅ RETURN NÄR DU VILL VETA SVARET";
470 INPUT Q$
480 PRINT:PRINT TAB(10);"SVARET =";R;" KRONOR"
490 PRINT:PRINT TAB(10)"TRYCK PÅ RETURN";:INPUT Q$
500 GOTO 10
510 CLS
520 PRINT:PRINT TAB(10);"HÄR ÄR DITT RÄKNEEXEMPEL"
530 PRINT:PRINT TAB(10);"KAPITAL =";K;" KRONOR"
540 PRINT:PRINT TAB(10)"PROCENTEN =";P;" %"
550 PRINT:PRINT TAB(10)"RÄNTA PÅ PERIODEN =";R;" KRONOR"
560 PRINT:PRINT TAB(10);"TRYCK PÅ RETURN NÄR DU VILL VETA SVARET";
570 INPUT Q$
580 PRINT:PRINT TAB(10);"SVARET =";T;
590 IF A=1 THEN PRINT "DAGAR" ELSE PRINT "MANADER"
600 PRINT:PRINT TAB(10)"TRYCK PÅ RETURN";:INPUT Q$
610 GOTO 10
620 REM DAGAR
630 T=INT(RND*330+30)
640 K=INT(RND*24500+500)
650 P=INT(RND*12+8)
660 R=K*P/100*T/360:R=INT(R*100+.5)/100
670 RETURN
680 REM MANADER
690 T=INT(RND*12+1)
700 K=INT(RND*24500+500)
710 P=INT(RND*12+8)
720 R=K*P/100*T/12:R=INT(R*100+.5)/100
730 RETURN
```

Program om geometriträning



Av Christian Lindeberg

Den här veckan skall jag fortsätta med program för träning i matematik. Tidningen kom knappt i brevlådan i lördags innan det ringde och Sven i Malmö undrade om jag inte kunde skriva ett program, som tränade på geometri. Sven går på högstadiet och tycker att geometri är svårt.

Det finns bara ett bra sätt att lära sig matematik och geometri i synnerhet. Träna, träna och träna. När det gäller geometri, så är de flesta dåliga därför att de orkar inte rita figurer till sina uppgifter. I geometri är det så att lösningen på problemet ser mycket enklare ut när man har en figur att titta på. Försök också att rita figurerna någorlunda i skala.

Programmet

Programmet jag har gjort tar upp kvadrat, rektangel, triangel och cirkel. Programmet är uppbyggt på dubbeltrutiner. Om du tittar i raderna 90 och 110 så ser du att först går programmet till subrutiner, som ligger på raderna 1000, 2000, 3000 och 4000. I dessa subrutiner hämtar datorn sina värden beroende på vad du tänker räkna på för figurer. När datorn hämtat sina värden kommer den tillbaka till rad 100 där den sedan hoppar till den typ av beräkning, som du har begärt. Det här är ett ganska bra sätt att arbeta på, där du använder samma val för två olika beslut. Man kan naturligtvis lägga hela beräkningen och framtagningen av värde i samma rutin. Alla rutinerna är uppbyggda på samma sätt. Subrutinerna plockar fram värdena genom att använda slumptalsgeneratorn och ta ett heltal mellan 1 och 30. Du kan själv ta andra gränser genom att ändra talet 30 till eget önskemål.

Variabel

I subrutinerna har jag valt en variabel, som är mycket dålig och det är bokstaven O, som mycket lätt kan förväxlas med siffran 0. En del datormärken kan inte använda flera tecken till sina variabler, därför har jag bara använt ett tecken. Om du kan använda flera tecken, skriv BREDD i stället för B, LANGD i stället för L, HOJD i stället för H, OMK i stället för O, AREA i stället för A, BAS i stället för B i triangelfallet.

Du lägger märke till att jag inte skriver ÅÄÖ i variabelnamn. Datorn klarar inte av ÅÄÖ i variabelnamn eftersom dessa tecken inte betyder bokstäver för datorn utan andra symboler.

Det är på det viset att datorn arbetar med ett grundsystem, som man inte ändrar när man sätter till svenska tecken. I grundsystemet kan datorn bara arbeta med bokstäverna A till Z. Dessa är de enda som datorn accepterar som variabler.

Ej ÅÄÖ

Nu kanske du säger: Men tecknen för ÅÄÖ syns på skärmen och då måste jag kunna använda dem också. Svaret är nej eftersom datorn arbetar med enbart siffror och det du ser på skärmen är inte vad som händer i datorns minne. Datorn arbetar endast med de sifferkoder för variabler, som är godkända i grundsystemet.

Det du ser på skärmen är att när datorn till exempel skickar en signal att datorn skall skriva ut koden 65, går denna först till en teckengenerator för att översättas till ett grafiktecken, som skall skrivas ut på skärmen. I detta fall är grafiktecknet ett A. Hade datorn sänt koden 66 hade det blivit ett B på skärmen.

Alla dessa koder kan du hitta i din instruktionsbok. Slå upp benämningen ASCII, så får du alla sifferkoderna datorn använder för de olika tecknen.

Jag hoppas att du skall ha nytta av geometriprogrammet. Jag kommer eventuellt att fortsätta med flera sådana här program om ni så önskar.

```
10 CLS
20 PRINT:PRINT"     MENY"
30 PRINT:PRINT"1. KVADRAT"
40 PRINT:PRINT"2. REKTANGEL"
50 PRINT:PRINT"3. TRIANGEL"
60 PRINT:PRINT"4. CIRKEL"
70 PRINT:PRINT"5. SLUTA"
80 PRINT:PRINT"VAD VILL DU RÄKNA PÅ";:INPUT Q$
90 ON VAL(Q$) GOSUB 1000,2000,3000,4000,5000
100 ON VAL(Q$) GOTO 200,300,400,500
110 GOTO 10
200 CLS:PRINT:PRINT"KVADRAT"
210 PRINT:PRINT"VAD BLIR AREAN OM SIDAN=";B
220 PRINT:PRINT"AREAN=";:INPUT X
230 IF A=X THEN PRINT"RÄTT"::GOTO 250
240 PRINT:PRINT"FEL FöRSöK IGEN";:GOTO 210
250 PRINT:PRINT"OMKRETSEN=";:INPUT Y
260 IF O=Y THEN PRINT"RÄTT":GOTO 280
270 PRINT:PRINT"FEL FöRSöK IGEN";:GOTO 250
280 PRINT:PRINT"TRYCK PÅ RETURN";:INPUT W$
290 GOTO 10
300 CLS:PRINT:PRINT"REKTANGEL"
310 PRINT:PRINT"VAD BLIR AREAN OM BREDDEN=";B
315 PRINT:PRINT"OCH LÄNGDEN=";L
320 PRINT:PRINT"AREAN=";:INPUT X
330 IF A=X THEN PRINT"RÄTT"::GOTO 350
340 PRINT:PRINT"FEL FöRSöK IGEN";:GOTO 310
350 PRINT:PRINT"OMKRETSEN=";:INPUT Y
360 IF O=Y THEN PRINT"RÄTT":GOTO 380
370 PRINT:PRINT"FEL FöRSöK IGEN";:GOTO 350
380 PRINT:PRINT"TRYCK PÅ RETURN";:INPUT W$
390 GOTO 10
400 CLS:PRINT:PRINT"TRIANGEL"
410 PRINT:PRINT"VAD BLIR AREAN OM BASEN=";B
415 PRINT:PRINT"OCH HöJDEN=";H
420 PRINT:PRINT"AREAN=";:INPUT X
430 IF A=X THEN PRINT"RÄTT"::GOTO 480
440 PRINT:PRINT"FEL FöRSöK IGEN";:GOTO 410
480 PRINT:PRINT"TRYCK PÅ RETURN";:INPUT W$
490 GOTO 10
500 CLS:PRINT:PRINT"CIRKEL"
510 PRINT:PRINT"VAD BLIR AREAN OM RADIEN=";R
515 PRINT:PRINT"PI=3.14"
520 PRINT:PRINT"AREAN=";:INPUT X
530 IF A=X THEN PRINT"RÄTT"::GOTO 550
540 PRINT:PRINT"FEL FöRSöK IGEN";:GOTO 510
550 PRINT:PRINT"OMKRETSEN=";:INPUT Y
560 IF O=Y THEN PRINT"RÄTT":GOTO 580
570 PRINT:PRINT"FEL FöRSöK IGEN";:GOTO 550
580 PRINT:PRINT"TRYCK PÅ RETURN";:INPUT W$
590 GOTO 10
1000 REM KVADRAT
1010 B=INT(RND*30)+1
1020 REM
1030 A=B*B
1040 O=4*B
1050 RETURN
2000 REM REKTANGEL
2010 B=INT(RND*30)+1
2020 L=INT(RND*30)+1
2025 IF B=L THEN 2010
2030 A=B*L
2040 O=2*B+2*L
2050 RETURN
3000 REM TRIANGEL
3010 B=INT(RND*30)+1
3020 H=INT(RND*30)+1
3025 IF B=L THEN 2010
3030 A=B*H/2
3050 RETURN
4000 REM CIRKEL
4010 R=INT(RND*30)+1
4020 REM
4030 A=R*R*3.14
4040 O=2*R*3.14
4050 RETURN
5000 END
```

Program för OS



Av Christian Lindeberg

Nu har OS i Calgary börjat och de sena kvällarna. Jag fick en fråga, som tyvärr kom lite för sent för förra numret. Det gällde program för inskrivning och sortering av resultat.

Först skall jag ta upp ett brev, som jag fick här i veckan från Rudi D. Jag skall ta ett utdrag ur brevet eftersom det här gäller ett problem, som ofta dyker upp.

"Jag har en Commodore plus/4, en VIC 20+exp 8K, en Commodore 64 'personal computer', kassettstationer Commodore 1530 och 1531 samt en skrivare Citizen 120D.

Om 64:an inte är en hemdator utan faktiskt en 'PC-dator', är mitt problem kanske enklare att lösa. Problemet är att kunna använda plus/4:ans ordbehandlare och helst även den övriga inbyggda mjukvaran på ett sådant sätt att jag kan lagra på disketter som sedan kan läsas av IBM-kompatibla maskiner. I din artikel från 26/9-87 ger du några basicsatser för att kunna lagra i ASCII-format. Jag har ingen diskdrive än men skall nu köpa en och jag tror att Commodore 1571 kunde passa bra. Kan jag i så fall använda dessa satser, både på ordbehandling och basicprogram och hur då?"

Problemen med datorer är att de både kan kommunicera med andra datorer och att de inte kan.

Låt mig dela upp problemet i delar. En dator har en intern kommunikation och en extern kommunikation. Med intern kommunikation menar jag kommunikation mellan datorns olika enheter, såsom diskdrive, skrivare och inom datorn. Den externa kommunikationen sker när datorn kommunicerar med en annan dator via någon form av datakommunikation.

Om vi tar exemplet med en 64:a och en diskdrive, så har diskdriven en egen dator, som styr systemet och uppläggningen av programmen på disketten. En diskett, som innehåller program eller filer av vilket slag som helst kan inte användas på en IBM-kompatibel dator. Systemen har olika sätt att hantera informationen varför det inte är möjligt att göra.

Finns det då ingen möjlighet att med hjälp av en till exempel 64:a eller plus/4 överföra filer från den till en IBM-kompatibel dator. Svaret är ja.

Vad som behövs är att båda datorerna har ett kommunikationsprogram och ett modem. Har man detta så är det bara att ringa upp den andra datorn och skicka över sin text och den andra datorn kan lagra texten på diskett. Kommunikationen kan ske i båda riktningarna utan problem. Ett bra modem i dag kostar kring 1.000 kr till 1.500 kr och ett kommunikationsprogram kan man få gratis via en del av alla de databaser, som finns i Sverige. Runt Malmö finns det fyra stycken som har kommunikationsprogram. Det enda problemet är hönan och ägget — vad — kom först? Eftersom man måste ha ett kommunikationsprogram, för att kunna hämta ett ur en databas. Men det finns säkert någon kompis i närheten som kan hjälpa dig med ett program.

Att ha ett modem och kommunikationsprogram vidgar också vyerna eftersom man kommer i kontakt med så många likasinnade i de olika databaserna. I dessa kan man också skicka brev och ställa frågor.

Till slut till Rudi: Din sista fråga är om du skall köpa en diskdrive 1571. Svaret är absolut nej eftersom den är speciellt gjord till C-128 och inte till dina datorer. Det är bara en liten del av 1571:an du kan utnyttja till dina datorer. Är det en begagnad till ett lågt pris så är det OK, men inte en ny.

Nu över till programmet. Jag har gjort ett litet program med en enkel sortering.

Rad 10 dimensionerar variablerna till 25 tävlande. Vill du ha fler så bara öka 25 till ett annat tal.

Rad 30 räknar antalet tävlande.

Raderna 40 till 70 — här sker inmatning av de tävlandes namn och tider.

Rad 80 hoppar tillbaka till rad 30 om det är första tävlande eftersom det inte är lönt att sortera en tävlande.

Raderna 90 till 150 — här sker sortering av de tävlande så att den med lagsta resultatet blir rankad etta. Om du vill ha det högsta resultatet först så vänd på tecknet i rad 110, så att det blir större än-tecknet.

Raderna 160 till 190 skriver ut resultatet.

Raderna 210 och 220 ger ett hopp tillbaka till nästa tävlande. Man kan komplettera programmet med en möjlighet att skriva in två resultat, så att det passar till de olika formerna av utförsåkning. Lycka till med programmet och ha ett skönt sportlov.

```
10 DIM N$(25),T(25)
20 CLS
30 R=R+1
40 PRINT:PRINT"MATA IN NAMN";
50 INPUT N$(R)
60 PRINT:PRINT"MATA IN TID ELLER POÄNG";
70 INPUT T(R)
80 IF R=1 THEN 30
90 FOR I=1 TO R-1
100 FOR J=I+1 TO R
110 IF T(I)<T(J) THEN 140
120 X=T(I):T(I)=T(J):T(J)=X
130 X$=N$(I):N$(I)=N$(J):N$(J)=X$
140 NEXT J
150 NEXT I
160 PRINT:PRINT:PRINT
170 FOR I=1 TO R
180 PRINT N$(I),T(I)
190 NEXT I
200 PRINT:PRINT
210 INPUT"TRYCK PÅ RETURN";Q$
220 GOTO 20
```

Så kollar datorn ditt personnummer



Hem DATORER

Av Christian Lindeberg

Vad betyder siffrorna efter födelsetiden i våra personnummer? Alltså de fyra sista siffrorna? Det är ett par frågor som återkommer med jämna mellanrum. Så här ligger det till:

Före 1947 hade vi inga personnummer eller liknande. Det enda vi använde var födelsetiden. 1 januari 1947 stadgades genom folkbokföringsförordningen att alla personer, som var mantalsskrivna i Sverige skulle få ett födelsenummer i det län där de var mantalsskrivna. Detta får en viss betydelse när du använder dagens dataprogram. Efter 1 januari 1947 har alla nyfödda fått ett födelsenummer från det län där de är födda. Den sista siffran i födelsenumret är udda för män och jämn för kvinnor.



Infördes 1967

9 juni 1967 kom ett tillägg till folkbokföringsförordningen att varje person, som är mantalsskriven i Sverige skall ha en identitetsbeteckning, som kallas personnummer. Detta personnummer skulle bestå av tre delar.

1. Födelsetid i ordningen de två sista siffrorna i året, de två ordningssiffrorna för månad och till slut de två siffrorna för dagen.
2. Födelsenummer från det län där man var mantalsskriven 1 jan 1947 eller född efter den 1/1 1947.
3. Kontrollsiffra.

Födelsenummer upp till 929 är utdelade till alla län i Sverige.

Alla födelsenummer mellan 930 och 999 tillhör Riksskatteverket och delas ut till länen om deras egna tar slut. Alla, som kommer till Sverige från utlandet får också ett födelsenummer mellan 930 och 999.

Subrutin

Kring allt det här har jag nu gjort ett program, som kanske fordrar en förklaring.

Jag börjar med att dimensionera indexerade variabler, som blir större än tio.

I rad 20 hoppar daton till en subrutin, som börjar i rad 340.

Här läser datorn in alla uppgifter om vilket län och vilka intervall, som födelsenumren ligger mellan. I variabeln N$(I) ligger länets namn i T(I) och R(I) ligger lägsta respektive högsta värdet på födelsenumret.

När subrutinen är inläst frågar datorn efter personnumret.



Detta måste skrivas, som födelsetid, bindestreck och sedan födelsenummer och kontrollsiffra i en följd. Mitt personnummer blir då 701216—1234.



● *Redan i livets allra första timme börjar "numreringen" på BB — och några veckor senare har den nyfödda ett personnummer som räcker livet ut.*

I raderna 60 till 180 räknar datorn ut och kontrollerar personnumret, om kontrollsiffran inte stämmer, så skickar datorn dig tillbaka till rad 40.

Uträkning av kontrollsiffra hade jag med för några veckor sedan, så det tar jag inte med nu.

Testar

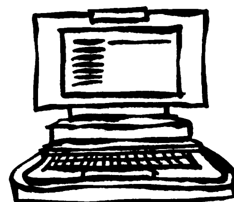
Rad 190 tar ut födelsenumret och lägger det i variabeln F.

I rad 200 till 220 testar datorn inom vilka gränser födelsenumret ligger och när detta är klart hoppar datorn till rad 230.

I raderna 230 till 330 skriver datorn ut den information, som den har plockat ur personnumret.

I raderna 270 och 280 testar datorn udda och jämnt.

Rad 290 testar om talet är större än 930 för då är det ett födelsenummer från Riksskatteverket.



Nu skulle man kunna hoppa från rad 330 till början av programmet och testa ett nytt personnummer.

Skriv en rad 330 som fungerar. Du kan börja med GOTO 30.

Sedan kan du sätta igång och tänka. Lycka till!

```
10 DIM N$(25),T(25),R(25)
20 GOSUB 340
30 CLS
40 PRINT:PRINT"SLA IN DITT PESONNUMMER MED BINDESTRECK";
50 INPUT A$
60 B$=MID$(A$,1,6)+MID$(A$,8,3)
70 FOR I=1 TO 9
80 A=VAL(MID$(B$,I,1))
90 IF I/2=INT(I/2) THEN B=B+A:GOTO 130
100 C=A*2
110 IF C>9 THEN C=C-9
120 B=B+C
130 NEXT I
140 D$=STR$(B)
150 E=VAL(MID$(D$,3,1))
160 E=10-E
170 IF E=VAL(RIGHT$(A$,1)) THEN 190
180 PRINT"PERSONNUMMER FEL":GOTO 40
190 F=VAL(MID$(A$,8,3))
200 FOR I=1 TO 25
210 IF F>T(I) AND F<R(I) THEN 230
220 NEXT I
230 CLS
240 PRINT:PRINT"DITT PERSONNUMMER= ";A$
250 PRINT:PRINT"DIN KONTROLLSIFFRA =";E
260 PRINT:PRINT"DITT FöDELSENUMMER= ";F
270 IF VAL(MID$(A$,10,1))/2=INT(VAL(MID$(A$,10,1))/2)
    THEN PRINT:PRINT"DU ÄR KVINNA"
280 IF VAL(MID$(A$,10,1))/2<>INT(VAL(MID$(A$,10,1))/2)
    THEN PRINT:PRINT"DU ÄR MAN"
290 IF F<930 THEN PRINT:PRINT"DU ÄR FöDD I ";N$(I):END
300 PRINT:PRINT"DITT LÄNS FöDELSENUMMER RÄCKTE INTE TILL"
310 PRINT:PRINT"SÅ ATT MAN FICK LÅNA AV RIKSSKATTEVERKET"
320 PRINT:PRINT"ELLER SÅ ÄR DU INFLYTTAD FRÅN UTLANDET"
330 END
340 FOR I=1 TO 25
350 READ N$(I),T(I),R(I)
360 NEXT I
370 RETURN
380 DATA STOCKHOLMS LÄN,001,139
390 DATA UPPSALA LÄN,140,159
400 DATA SöDERMANLANDS LÄN,160,189
410 DATA öSTERGöTLANDS LÄN,190,239
420 DATA JöNKöPINGS LÄN,240,269
430 DATA KRONOBERGS LÄN,270,289
440 DATA KALMAR LÄN,290,310
450 DATA GOTLANDS LÄN,320,329
460 DATA BLEKINGE LÄN,330,349
470 DATA KRISTIANDSTADS LÄN,350,389
480 DATA MALMöHUS LÄN,390,459
490 DATA HALLANDS LÄN,460,479
500 DATA GöTEBORGS OCH BOHUS LÄN,480,549
510 DATA ÄLVSBORGS LÄN,550,589
520 DATA SKARABORGS LÄN,590,619
530 DATA VÄRMLANDS LÄN,620,659
540 DATA öREBRO LÄN,660,689
550 DATA VÄSTMANLANDS LÄN,690,709
560 DATA KOPPARBERGS LÄN,710,749
570 DATA GÄVLEBORGS LÄN,750,779
580 DATA VÄSTERNORRLANDS LÄN,780,819
590 DATA JÄMTLANDS LÄN,820,849
600 DATA VÄSTERBOTTENS LÄN,850,889
610 DATA NORRBOTTENS LÄN,890,929
620 DATA RIKSSKATTEVERKET,930,999
```

Fibonaccis tal och kaninproblem

Den här veckan kommer det bland annat att handla om matematik. Innan jag kommer till den matematiska delen så skall jag ta en brevfråga om kopiering.



Av Christian Lindeberg

Brevet är från Peter B i Hästveda, som bland annat skriver så här:

"Jag har en Spektravideo 738 med inbyggd diskettstation. Mitt problem är nu att jag vill kopiera in spel på diskett. Jag kan kopiera in sådana spel som går att lista upp men inte sådana spel som inte går att lista upp bland annat sådana spel som man laddar in med BLOAD "CAS:",R. Jag har också hört att man kan kopiera in cartridges på diskett. Vet du hur man gör?"

Skyddat

Jag tror att många känner igen sig i frågeställningen. Nu är det så att de företag, som har gjort spelen har lagt ett kopieringsskydd för att spelen inte skall gå att kopiera. Jag hoppas att du vet att det inte är tillåtet att kopiera köpta spel och kopiera till kompisar. Du får däremot kopiera till dig själv, så att du inte sliter på orginalspelet. För att kunna kopiera ett spel, som är skyddat så måste man ha speciella kopieringsprogram, eftersom spelen har AUTOSTART. Du måste ladda in ett program i datorn för att kunna kopiera det till disketten.



Via telefon

När du laddar i programmet så startar det upp och du kommer inte in i datorns minne, så att du kan kopiera det. Det finns på marknaden speciella kopieringsprogram för att kopiera från band till disk och från disk till disk. Nu är din dator inte en av de vanligaste på marknaden, så jag vet inte om det finns sådana program i Sverige. Titta efter i svenska eller engelska datortidningar. Ett annat sätt att kringgå det är om man har ett modem och terminalprogram och en kompis med samma dator också med modem och terminalprogram, så kan man kanske föra över spelet via telefon till diskettenheten.

Primtal

Nu över till matematiken. Jag fick två frågor i veckan, den ena gällde primtal och den andra Fibonaccital. Först tar vi primtal. Primtal är ett heltal, som inte har någon annan delare än 1 och sig självt. Det betyder att ett primtal inte kan delas upp i andra tal. Det finns bara ett jämnt primtal och det är talet 2. De första primtalen är 1, 2, 3, 5, 7, 11 och så vidare. I programmet till i dag har jag gjort en liten lopp, som tar ut alla primtal fram till 500. Kör programmet och fundera ut varför det tar längre och längre tid innan nästa primtal kommer fram. Kan man sina primtal så är det mycket enklare att ta fram minsta gemensamma nämnare vid bråkräkning.

Kaninproblemet

Fibonaccital eller Fibonaccis följd är en oändlig följd av heltal. Först skall jag tala om vem Fibonacci var. Leonardo av Pisa, som han också kallades var en matematiker från 1200-talet. Matematiken utvecklades mycket långsamt under medeltiden och Fibonacci var en av de få.



Han gav ut en bok år 1202 med titeln "Liber abaci" (En bok om abacus). En abacus är en ram eller ett bräde för underlättande av räkneoperationer. Detta system användes fortfarande i Japan bland annat. Fibonacci bidrog till att de hindu-arabiska siffrorna kom till Europa, alltså de siffror, som användes i dag.

Fibonaccis tal sammanhänger med det s k "kaninproblemet". Man placerar ett kaninpar i en stor bur för att ta reda på hur många kaniner det blir på ett år eller längre.

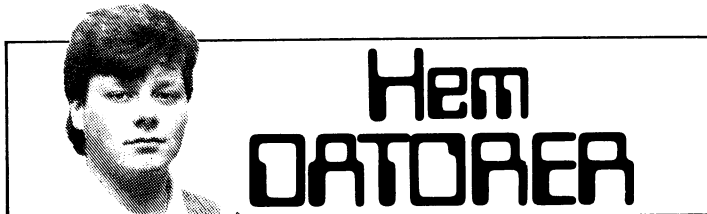
Oändlig följd

Kaninparet föder ett nytt par varje månad, från och med den andra månaden av sitt liv. Det första paret föder ett par första månaden så att efter första månadens slut finns det två par. Av dessa två par föder ett par ett nytt par av kaniner men det andra paret får vänta en månad till innan de kan producera nya kaniner. Vid andra månadens slut finns det alltså 3 (=1+2) kaninpar och vid tredje månadens slut finns det 5 (=2+3) kaninpar. Kommande månad blir det då 8 (=3+5). Om vi tittar på talföljden så ser vi något intressant. 1, 2, 3, 5 och 8, vad kommer nästa att bli? Jo, lägg ihop de två sista 8 och 5 så får du 13, vilket är nästa tal. Fibonacci följden har följande definition: Det är en oändlig följd av tal, där de två första talen är 0 och 1 och varje tal i fortsättningen är summan av de två föregående. Det blir alltså 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 o s v.

I programmet, som jag har gjort kan du se både primtal och Fibonaccital. Hur många kaniner blir det på ett år eller två?

```
10 CLS
20 PRINT:PRINT"MENY"
30 PRINT:PRINT"1. PRIMTAL"
40 PRINT:PRINT"2. FIBONACCITA
50 PRINT:PRINT"3. AVSLUTA"
60 PRINT:PRINT"VAD öNSKAS";
70 INPUT A
80 IF A<1 OR A>3 THEN 10
90 ON A GOTO 100,250,470
100 REM PRIMTAL
110 CLS
120 PRINT"PRIMTAL"
130 PRINT"*******"
140 PRINT
150 PRINT"2";
160 FOR I=3 TO 500 STEP 2
170 FOR J=3 TO I/2
180 IF I/J=INT(I/J) THEN 210
190 NEXT J
200 PRINT",";I;
210 NEXT I
220 PRINT:PRINT:PRINT
230 INPUT "TRYCK PA RETURN NAR DU AR KLAR";Q$
240 GOTO 10
250 REM FIBONACCITAL
260 CLS
270 PRINT"FIBONACCITAL"
280 PRINT"************"
290 PRINT
300 A=1
310 B=2
320 PRINT A;TAB(15);"MANAD 0"
330 PRINT B;TAB(15);"MANAD 1"
340 N=1
350 A=A+B:N=N+1
360 IF A>100000! THEN 440
370 PRINT A;TAB(15);"MANAD";N
380 IF INT(N/12)=N/12 THEN PRINT:PRINT:INPUT Q$
390 B=A+B:N=N+1
400 IF B>100000! THEN 440
410 PRINT B;TAB(15);"MANAD";N
420 IF INT(N/12)=N/12 THEN PRINT:PRINT:INPUT Q$
430 GOTO 350
440 PRINT:PRINT:PRINT
450 INPUT "TRYCK PA RETURN NAR DU AR KLAR";Q$
460 GOTO 10
470 END
```

Hur göra för att få fler decimaler?



Av Christian Lindeberg

För en vecka sedan hade jag en artikel om primtal, där programmen blev något otydliga. Jag skall ta med det intressantaste den här gången också.

En av de som ringde i lördags undrade över hur man gjorde för att datorn skulle räkna med flera decimaler vid en division. Alla har gjort erfarenheten att datorn ger inte mer än 6 eller 8 decimaler. Har du en PC kan du få upp till 16 decimaler. Hur skall man då göra för att komma fram till flera decimaler?

Så gör du

När man skall lösa ett problem så måste man dela upp det i sina beståndsdelar. Ta ett papper och en penna och sätt igång och skriv. Vi tar talen 47/8 om vi gör första divisionen så får vi att första siffran blir 5 och vi får en rest på 7. Efter 5 skall det vara ett kommatecken. Nu har vi 7/8 kvar. Vi multiplicerar 7 med 10 och får då 70/8, vilket blir 8. Detta är första decimalen, 8*8=64 och 70−64=6. Denna siffra multiplicerar vi med 10 igen och får då 60/8 vilket blir 7, som blir 2:a decimalen. 7*8=56 och drar vi detta ifrån 60 får vi 4, som vi multiplicerar med 10. Nu kommer divisionen med 8 och vi får 40/8, som blir 5 och ingen rest. Vi har nu fått tre decimaler och en heltalssiffra. Svaret blir alltså 5,875.

Och så datorn

Om vi nu lämnar papper och penna och slår på datorn, så skall vi skriva ner ett program för att lösa uppgiften. Vi börjar med att i rad 10 tömma skärmen med ett lämpligt kommando CLS eller vad som gäller för din dator. Sedan bör vi ha några förklarande PRINT-satser till exempel:

```
20 PRINT:PRINT "VILKA TAL VILL DU DIVIDERA"
30 PRINT:PRINT "SLÅ IN TALEN MED KOMMATECKEN MELLAN TALEN";
```

Genom att slå in talen med ett kommatecken mellan talen så kan jag använda en INPUT-sats, så att rad 40 blir:

```
40 INPUT T,N
```

Jag använder T för täljare och N för nämnare. I rad 50 behöver vi en förklarande PRINT-sats.

```
50 PRINT:PRINT "HUR MÅNGA DECIMALER VILL DU HA";
```

I rad 60 lägger vi en INPUT-sats med variabeln D för att ta hand om antalet decimaler.

```
60 INPUT D
```

Nu har vi klarat av hela inmatningsbiten och kan gå över till beräkningsdelen. Eftersom vi har antalet decimaler i en variabel D så kan vi använda denna i en loop, som börjar så här:

```
70 FOR I=1 TO D
```

Heltalsdelen

Nu skall vi först plocka fram heltalsdelen av divisionen. Jag hoppas att du kommer ihåg en funktion, som heter INT och lämpar sig mycket bra för vad vi skall göra.

I rad 80 skriver vi A=INT(T/N). I variabeln A får vi nu resultatet av T/N där datorn har tagit bort decimalerna. I vårt exempel med 47/8 blir siffran 5.

I rad 90 låter vi datorn skriva ut variabeln A.

```
90 PRINT A;
```

Semikolon efter A är till för att resten av siffrorna skall komma att skrivas åt vänster efter varandra.

Nu är heltalsdelen klar, men eftersom vi skall använda loopen flera gånger, så måste vi sätta ut ett kommatecken efter första divisionen, och detta gör vi i rad 100.

```
100 IF I=1 THEN PRINT ",";
```

I nästa steg skall vi ta bort A*N ifrån T. I vårt exempel var A första gången 5 och N=8 och multiplicerar vi får vi 5*8=40.

I rad 110 skriver du följande:

```
110 T=T−A*N
```

Nästa steg

Nu kommer vi till nästa steg som blir att multiplicera resten med 10. Resten är T och nu multiplicerar vi T med 10 och får då i rad 120:

```
120 T=T*10
```

Nu är vi klara att börja om det hela en gång till och därför skriver vi i rad 130:

```
130 NEXT I
```

Nu hoppar datorn upp till rad 70 och gör samma saker igen, med undantag av rad 100, som den hoppar över eftersom variabeln I nu har blivit 2. Hela programmet blir så här:

```
10 CLS
20 PRINT:PRINT "VILKA TAL VILL DU DIVIDERA"
30 PRINT:PRINT "SLÅ IN TALEN MED KOMMATECKEN MELLAN TALEN";
40 INPUT T,N
50 PRINT:PRINT "HUR MÅNGA DECIMALER VILL DU HA";
60 INPUT D
70 FOR I=1 TO D
80 A=INT(T/N)
90 PRINT A;
100 IF I=1 THEN PRINT ",";
110 T=T−A*N
120 T=T*10
130 NEXT I
```

Vill du ha många beräkningar så kan du lägga till ett par rader i slutet.

```
140 PRINT:PRINT "VILL DU HA FLER BERÄKNINGAR (J/N)";
150 INPUT Q$
160 IF Q$="J" OR Q$="j" THEN 10
```

Nu känner datorn av om du har svarat JA med stor J eller med litet i och hoppar då till rad 10 för ny beräkning. I alla andra fall så avslutas programmet.

Lycka till!

En repris

Här kommer nu det snabba primtalsprogrmmet från förra veckan i repris:

```
10 CLS
20 PRINT"TALET";:INPUT A
30 GOTO 70
40 PRINT 2;
50 A=A/2
60 IF A=1 THEN 180
70 IF (A/2−INT(A/2))<>0 THEN B=3:GOTO 90
80 GOTO 40
90 C=SQR(A)+1
100 IF B>=C THEN 170
110 IF A/B−INT(A/B)=0 THEN 140
120 B=B+2
130 GOTO 100
140 PRINT B;
150 A=A/B
160 GOTO 90
170 PRINT A;
180 END
```

Så använder du indexerade variabler



Hem DATORER

Av Christian Lindeberg

De nästa två veckorna kommer jag att skriva om indexerade variabler, som ibland ställer till problem. Det kommer många frågor om dessa typer av variabler.

För det första så använder man två olika namn på samma sak antingen indicerade variabler eller indexerade variabler. I fortsättningen kommer jag att använda uttrycket indexerade variabler.

När vi använder variabler så tar vi bokstäver och ord ur vårt alfabet, till exempel A, B, C o s v. Om jag nu vill ha åtta olika variabler så kan jag naturligtvis använda mig av A, B, C, D, E, F, G och H, som variabler. Nackdelen är att jag måste ha åtta Input-satser.

```
10 Input A
20 Input B
30 Input C
o s v.
```

Detta tar onödigt mycket utrymme. Jag skulle kunna använda mig av en indexerad variabel i stället. En indexerad variabel består av en bokstav eller ett ord och ett index inom parentes. I exemplen kommer jag att använda mig av enstaka bokstäver, eftersom en del datorer inte klarar av ord, som variabler.

Jag gör nu åtta variabler med hjälp av A och ett index och då får jag:
A(1), A(2), A(3), A(4), A(5), A(6), A(7) och A(8).

Skillnaden

Vad blir nu den stora skillnaden? Jo, att nu kan jag använda en loop för att få siffrorna till mina indexerade variabler. Så i stället för åtta rader med Input så får jag följande konstruktion.

```
10 For I = 1 To 8
20 Input A(1)
30 Next I
```

Nu har jag på tre rader klarat av mina åtta variabler. Vill jag nu ha fler så ökar jag i rad 10 till det tal jag önskar. Men om du vill ha fler än 10 så måste datorn veta av detta annars blir den sur och skriver "Out of dimension error" eller "Subscript out of range". Normalt reserverar datorn plats för 0 till 10, som indexerade variabler och skall man över detta, så måste datorn i början av programmet ha en dimensionering och detta gör vi genom att skriva till exempel:

```
5 Dim A(20)
```

om det är 20 variabler, som vi tänker oss att använda.

Betygshistogram

Jag skall visa med ett litet exempel hur man använder indexerade variabler. Jag skall göra ett program, som i ett liggande histogram visar fördelning över till betyg.

Först måste vi ha en inmatningsloop.

```
10 CLS
20 For I = 1 To 5
30 Input A(I)
40 Next I
```

I loopen använder jag indexet från 1 till 5, som betecknar betygen. I variablen A(I) skall jag nu lägga hur många, som har de olika betygen.

Nu behöver jag en loop, som skriver ut det liggande staplarna.

Ja, i själva verket så behöver jag två loopar som är nestade, vilket betyder att den ena loopen ligger inuti den andra på följande sätt:

```
50 CLS
60 For I = 1 To 5
70 Print I; " ":
80 For J = 1 To A(I)
90 Print "*":
100 Next J
110 Print: Print
120 Next I
```

Värdena

Den yttersta loopen är I och går från 1 till 5. Den innersta loopen är J och går från 1 till A(I), som är antalet betyg. Kom ihåg att nestade loopar aldrig får korsa en annan loop. I exemplet här ser du att looparna kommer på följande sätt:

```
For I = 1 To 5
For J = 1 To A(I)
Next J
Next I
```

Den innersta loopen måste vara avslutad innan nästa loop's index ökar med ett.

Hur många nestade loopar man kan använda beror på datorn.

Efter staplarna vill jag nu skriva ut värdena eftersom jag har en CLS (tömmer skärmen) i rad 50, som tar bort de inslagna värdena.

För att få värdena under betygen så använder jag mig av Tab-funktionen på följande sätt:

```
130 Print: Print: Print
140 For I = 1 To 5
150 Print Tab (I*5): I:
160 Next I
170 Print
180 For I = 1 To 5
190 Print Tab (I*5): A(I):
200 Next I
```

Lagom avstånd

Genom att göra två loopar och använda index I i Tab multiplicerat med 5, så får jag lagom långa avstånd mellan siffrorna. Försök med andra tal än 5 för att se hur det ändrar sig på din skärm. Hela programmet blir nu:

```
10 CLS
20 For I = 1 To 5
30 Input A(I)
40 Next I
50 CLS
60 For I = 1 To 5
70 Print I: " ":
80 For J = 1 To A(I)
90 Print "*":
100 Next J
110 Print: Print
120 Next I
130 Print: Print: Print
140 For I = 1 To 5
150 Print Tab (I*5): I:
160 Next I
170 Print
180 For I = 1 To 5
190 Print Tab (*5): A(I):
200 Next I
```

Ändra nu gärna variablernas storlek men kom ihåg att du måste dimensionera när du kommer över 10.

Nästa vecka är det påsk och då fortsätter jag med indexerade variabler och hur man använder dem i sortering.

Repris på lottorättningen



Hem DATORER

Av Christian Lindeberg

Den här veckan skall jag ta en repris på lottorättningen, som många haft stora problem med. Innan jag börjar med programmet skall jag komma med en upplysning.

Det är självklart att om någon vill ta upp ett problem, som jag kan skriva om, men inte vill att deras namn skall synas i tidningen, så bara skriv det i brevet. Det är en sak som jag naturligtvis respekterar att man vill vara anonym.

Endast 10 rader

Lottorättningsprogrammet hade en svaghet och det var att det inte fanns möjlighet att använda mer än 10 rader beroende på det sätt som jag hade dimensionerat programmet. Idag har jag gjort ett program, som hanterar 30 rader. För att undvika missförstånd så tar jag med hela programmet.

Låt oss börja med dimensioneringssatserna och inläsningen av DATA-satserna.

```
10 DIM N(30,7),R(30),T(30)
20 CLS
30 FOR I=1 TO 30
40 FOR J=1 TO 7
50 READ N(I,J)
60 NEXT J
70 NEXT I
```

I rad 10 ser du att det måste finnas tre dimensioneringar, men att man kan skriva dem på en rad med kommatecken mellan variablerna. I rad 20 så tömmer programmet skärmen.

Raderna 30 till 70 läser in DATA-satserna och lagrar dem i variabeln N(I,J), som är dimensionerad till 30 rader med 7 nummer i varje rad.

Inte läsa in

När datorn läser in DATA-satserna, så lagras dessa på ett speciellt ställe i minnet och man kan sedan inte läsa in dessa igen under samma körning. Om man av någon anledning skulle vilja läsa in DATA-satserna, igen så måste man lägga in en funktion, som heter RESTORE innan man kan göra en ny inläsning av samma DATA-satser.

Nu till nästa del av programmet.

```
80 CLS
90 PRINT:PRINT "VILKEN ÄR DEN RÄTTA RADEN?"
100 PRINT:PRINT "SLÅ IN RADEN NÄR FRÅGORNA KOMMER"
110 PRINT
120 FOR I=1 TO 7
130 PRINT "VILKET ÄR RÄTT NUMMER?";I;
140 INPUT A(I)
150 NEXT I
160 PRINT:PRINT"VILKET ÄR FÖRSTA TILLÄGGSNUMRET";
170 INPUT T1
180 PRINT:PRINT"VILKET ÄR ANDRA TILLÄGGSNUMRET?"
190 INPUT T2
```

Här begär datorn information om den rätta lottoraden och de två tilläggsnumren.

Nu kommer vi till den avdelningen där datorn skall rätta dina lottorader. Det är viktigt att alla variabler och summeringar blir riktiga. I rad 230 står det R(I)-R(I)+1 här räknar datorn samman antalet rätt för varje rad.

I rad 250 kontrollerar datorn om det finns några tilläggsnummer.

Slutet på rad 250 skall vara T(I)-T(I)+1.

"Skumma" fel

Tänk på att när ni skriver i basic så kan det hända att man får en del skumma fel om man inte är tillräckligt noggrann med syntaxen. I basic kan man slarva rätt mycket men ibland straffas man med en rad dunkla fel.

Kom ihåg att när ni skriver IF-satser så skall man inte ha mellanslag. Se rad 230 det skall inte vara:

IF A(K) - N(I,J) THEN R (I) - R (I) + 1, utan titta i rad 230 hur det skall stå. Använd inte fler mellanslag än nödvändigt.

```
200 FOR I=1 TO 30
210 FOR J=1 TO 7
220 FOR K=1 TO 7
230 IF A(K)=N(I,J) THEN R(I)=R(I)+1
240 NEXT K
250 IF T1=N(I,J) OR T2=N(I,J) THEN T(I)=T(I)+1
260 NEXT J
270 NEXT I
```

Tre loopar

Lägg märke till att jag har tre stycken loopar. Detta kallas för tre nästade loopar. Observera att looparna inte korsar varandra utan I-loopen är ytterst och sedan kommer J-loopen och allra innerst har vi K-loopen.

Nu har vi bara en sak kvar och det är utskriften och DATA-satserna, och här kommer de:

```
280 FOR I=1 TO 30
290 PRINT:PRINT"RAD NUMMER";I;"HADE";R(I);"RÄTT OCH";T(I)"TILLÄGGSNUMMER";
300 IF R(I)+T(I)>4 THEN PRINT TAB(50);"V I N S T";:GOTO 320
310 IF I/15=INT(I/15) THEN PRINT:PRINT :INPUT QQ$
320 NEXT I
330 DATA 2,4,6,12,23,34,35
340 DATA 4,7,9,10,15,18,19
350 DATA 7,9,12,15,17,21,25
360 DATA 5,9,13,16,31,32,34
370 DATA 12,14,16,17,19,21,34
380 DATA 13,15,21,24,26,32,35
390 DATA 21,24,26,27,29,31,34
400 DATA 23,25,28,29,32,33,35
410 DATA 12,14,15,19,21,25,26
420 DATA 1,5,8,14,19,21,27
430 DATA 1,4,6,12,23,34,35
440 DATA 4,7,8,10,15,18,19
450 DATA 7,9,11,15,17,21,25
460 DATA 5,9,12,16,31,32,34
470 DATA 12,14,16,17,19,21,34
3480 DATA 13,15,21,25,26,32,35
490 DATA 21,24,26,28,29,31,34
500 DATA 23,25,28,30,32,33,35
510 DATA 12,14,15,18,21,25,26
520 DATA 1,5,8,14,19,22,27
530 DATA 2,4,6,12,23,33,35
540 DATA 4,7,9,10,15,18,20
550 DATA 7,9,12,15,17,21,26
560 DATA 5,9,13,16,31,32,35
570 DATA 12,14,16,17,19,22,34
580 DATA 13,15,21,24,26,33,35
590 DATA 21,24,26,27,29,33,34
600 DATA 23,25,28,29,32,34,35
610 DATA 12,14,15,19,21,22,26
620 DATA 1,5,8,14,19,21,28
```

Tänk på att det är bara två lördagar kvar innan jag gör ett sommaruppehåll. Ha en bra vecka!

Så räkna fram idea

Tack för alla brev och samtal angående bantning och hur att räkna ut vikter. Det finns ett annat sätt att räkna ut idealvikten och detta fick jag i flera olika brev den gångna veckan.



Av Christian Lindeberg

En del hade också gjort program till sina förslag. Alla skrev om samma teknik nämligen att räkna ut ett BMI-värde. BMI betyder "Body Mass Index". Det var inget av programmen som tog hänsyn till att det är skillnad på män och kvinnor.

Jag hittade i Läkartidningen nummer 3/89 en artikel av docent Stephan Rössner och med dr Alicja Kuskowska-Wolk. Artikeln handlade om längd och vikt och där fanns också en tabell över BMI-kategorier.

Tabellen hittar du direkt under min vinjett är ovan.

BMI-kategorie(kg/m²)	Män	
Undervikt	BMI<20.1	,
Acceptabel vikt	BMI=20.1 - 25.0	BM
övervikt	BMI=25.1 - 30.0	BMI=.
Fetma	BMI>30.0	BMI>28.\

● *Tabell över BMI-kategorier — se vidare i artikelte.*

Räkna så

Hur räknar man då fram BMI?

Man tar vikten i kilogram och dividerar med kvadraten på längden i meter. Exempelvis om man väger 104 kg och är 1.88 meter blir BMI 104/(1.88★1.88)=29,4 vilket är för högt. Idealvikten för 1.88 m och man är mellan 20.1★1.88★1.88 och 25★1.88★1.88, vilket blir mellan 71,0 kg och 88,4 kg.

Jag har gjort ett program kring BMI, som räknar vad du skall väga och tar hänsyn till man och kvinna. Du hittar programmet här nedan. Det finns inga konstigheter i programmet. Den första delen tar reda på din vikt, längd och om du är man eller kvinna. Det finns två subrutiner en från rad 190 och en från rad 350. Den på rad 190 tar hand om utskriften för kvinnor.

Båda rutinerna är nästan identiska och skiljer sig bara på värdena.

Kilokalorier

Jag har fått påpekande från en del läsare att det heter kilokalorier och inte bara kalorier. Jag har använt talspråk när jag skriver och man talar bara om kalorier, men det är helt riktigt att den fysikaliska enheten är kilokalorier.

Ni som vill skicka program på diskett i stället för brev kan använda antingen PC-kompatibel diskett eller diskett till VIC-64 eller C-128.

Lycka till med viktprogrammet. Nästa vecka kommer ett program som räknar fram kilokalorieinnehållet i olika födoämnen.

```
10 CLS
20 PRINT :PRINT
30 PRINT TAB(10);"BERÄKNING AV DITT VIKTTILLSTAND"
40 PRINT :PRINT TAB(10);"SVARA PÅ FöLJANDE FRAGOR"
50 PRINT :PRINT TAB(10);"HUR MYCKET VÄGER DU I KG";
60 INPUT KG
70 PRINT :PRINT TAB(10);"HUR LANG ÄR DU SVARA I METER";
80 INPUT LD
90 PRINT :PRINT TAB(10)"ÄR DU MAN (M) ELLER KVINNA (K)";
100 INPUT S$
110 BMI=KG/(LD*LD)
120 IF S$="M" THEN GOSUB 190
130 IF S$="K" THEN GOSUB 350
140 PRINT :PRINT TAB(10);"VILL DU GöRA FLER BERÄNINGAR (J/N)";
150 INPUT Q$
160 IF Q$="J" OR Q$="j" THEN 10
170 PRINT :PRINT TAB(10);"TACK FöR DENNA GÅNG"
180 END
190 CLS
200 PRINT :PRINT
210 PRINT TAB(10);"BERÄKNING AV VIKTFAKTOR FöR MAN"
220 PRINT :PRINT TAB(10);"DITT BMI-TAL=";INT(BMI*10+.5)/10
230 PRINT :PRINT TAB(10);"NORMALVÄRDET LIGGER MELLAN 20.1 OCH 25.0"
240 PRINT :PRINT TAB(10);"DIN IDEALVIKT BöR LIGGA MELLAN";
250 PRINT INT(20.1*LD*LD*10+.5)/10;"KG OCH";
260 PRINT INT(25*LD*LD*10+.5)/10;"KG"
270 IF BMI<20.1 THEN V$="UNDERVIKT"
280 IF BMI<25 AND BMI>=20.1 THEN V$="ACCEPTABEL VIKT"
290 IF BMI<=30 AND BMI>=25.1 THEN V$="öVERVIKT"
300 IF BMI>30 THEN V$="FETMA"
310 PRINT :PRINT TAB(10)"DU HAR ";V$;
320 IF BMI>25.1 THEN PRINT " OCH BEHÖVER ANVÄNDA ETT BANTNINGSPROGRAM"
330 PRINT
340 RETURN
350 CLS
360 PRINT :PRINT
370 PRINT TAB(10);"BERÄKNING AV VIKTFAKTOR FöR KVINNA"
380 PRINT :PRINT TAB(10);"DITT BMI-TAL=";INT(BMI*10+.5)/10
390 PRINT :PRINT TAB(10);"NORMALVÄRDET LIGGER MELLAN 18.7 OCH 23.8"
400 PRINT :PRINT TAB(10);"DIN IDEALVIKT BöR LIGGA MELLAN";
410 PRINT INT(18.7*LD*LD*10+.5)/10;"KG OCH";
420 PRINT INT(23.8*LD*LD*10+.5)/10;"KG"
430 IF BMI<18.7 THEN V$="UNDERVIKT"
440 IF BMI<23.8 AND BMI>=18.7 THEN V$="ACCEPTABEL VIKT"
450 IF BMI<=28.6 AND BMI>=23.9 THEN V$="öVERVIKT"
460 IF BMI>28.6 THEN V$="FETMA"
470 PRINT :PRINT TAB(10)"DU HAR ";V$;
480 IF BMI>25.1 THEN PRINT " OCH BEHöVER ANVÄNDA ETT BANTNINGSPROGRAM"
490 PRINT
500 RETURN
```

Basicprogram till Amigan

Den här veckan skall jag ta upp en icke helt okänd dator, nämligen Amiga. Anledningen är att så många har skrivit och frågat om basicprogrammering till Amigan.



Hem DATORER

Av Christian Lindeberg

Amigan finns i tre olika modeller: 1000, som var den första Amigan, sedan kom Amiga 500 och 2000. När man skall ladda in basicen i datorn så är det skillnad på 1000 och de andra.

Om du har en Amiga 1000 så gör på följande sätt:

1. Sätt i Kickstart-disketten.
2. När nu ser Kickstart-symbolen så sätt i Workbench-disketten.
3. När du ser symbolerna för Workbench så sätt i Extras-disketten.

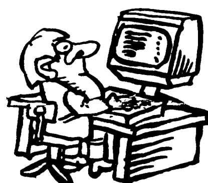
Om du har en Amiga 500 eller 2000, så slopa steg 1.

Amigans basic är från Microsoft och är i grunden samma basic som GWbasic och basicen till VIC-datorerna.

När man skall använda basicen på Amigan så måste man klicka fram menyn för Project, Edit, Run och Windows.

Där väljer du Windows och sätter i gång och skriver ner mina program eller dina egna.

Använd radnummer precis som jag gör i mina program.



Utan radnummer

Amiga-basicen kan mycket väl hantera program utan radnummer, men det stora problemet är att det är svårare att klara av det.

Om du behärskar Pascal, så kan du pröva på metoden. Har du läst "Starta med struktur" av Sven Burman och Bo Bäckman utgiven på Liber och fortsättningen "Fortsätt med struktur" av samma författare och förlag, så är du också mogen för att skriva program utan radnummer.

Men framför allt så läs igenom den instruktionsbok, som heter "Amiga Basic". Boken finns både på svenska och engelska.

Jag gjorde följande lilla program för att jämföra Amiga 500 med en vanlig PC med GWbasic.

```
10 CLS
20 RANDOMIZE(TIMER)
30 DIM A(1000)
40 N=300
50 FOR I=1 TO N
60 A(I)=INT(RND★1000)
70 NEXT I
80 FOR I=1 TO N-1
90 FOR J=I+1 TO N
100 IF A(I)<A(J) THEN 120
110 X=A(I):A(I)=A(J):A(J)=X
120 NEXT J
130 NEXT I
140 FOR I=1 TO N
150 PRINT A(I)
160 NEXT I
```

På PC-datorn tog ovanstående program sju minuter och sex sekunder.

När jag testade samma på Amigan trodde jag att den hade slutat fungera. Jag fick minska N till 40. För att ta ut 40 tal på Amigan och sedan sortera dem i stigande ordning så tog alltså Amigan sex minuter och tio sekunder. PC-datorn tog alltså sju minuter för att sortera 300 tal. Programmen är skrivna exakt lika på de båda datorerna.

Mycket långsam

Amigan är mycket långsam men har en stor fördel om man gillar ljud och grafik. Jag tycker att Amigan är den bästa datorn på marknaden när det gäller grafik och ljud.

Amiga-basicen innehåller naturligtvis många grafik- och ljudkommandon, jag skall bara ta upp ett ljudkommando som inte är så vanligt.

Kommandot, som egentligen är två heter SAY och TRANSLATE$. SAY kan användas separat men inte TRANSLATE$. Skriv följande på din Amiga:

SAY TRANSLATE$ ("Hi there, how are you?")

Internt engelsk

Datorn kommer nu att uttala ditt meddelande. Tyvärr så är datorn internt engelsk, så skall du skriva något som även låter bra på svenska får du pröva dig fram. Testa med följande:

SAY TRANSLATE$ ("Hur maar du ee dag?")

och jämför med den riktiga svenskan.

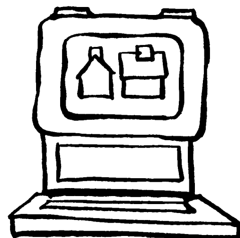
SAY TRANSLATE$ ("Hur mår du i dag?")

Amigan innehåller också en del bra kommandon för vanlig programmering. I exemplet med de 40 talen och sorteringen så hade man kunnat skriva raderna 100 och 110 som en rad så här:

```
100 IF A(I)>A(J) THEN SWAP A(I),A(J)
```

Rad 110 skall alltså bort, nu går programmet lite fortare. Nu tar det fem minuter och 23 sekunder för 40 tal.

Slutsats: skall du använda Amigan för sortering bör du skaffa dig ett snabbare sorteringsprogram.



Bra kommandon

Några andra bra kommandon är CHAIN, som kan hämta in ett annat basicprogram och i kombination med COMMON, som för över variabelvärdet till det nya programmet.

Det finns även MERGE som sätter ihop två program till ett. Tänk på att med MERGE så måste programmet sparas som ASCII-tecken.

Detta gör du genom att spara som SAVE "NAMN",A.

När du startar basic på Amigan så ser du att du har 25.000 bytes free. Detta är inte så mycket. Du kan utöka detta med kommandot CLEAR. Om du skriver CLEAR,45.000 så får du ett minne på 45.000 bytes.

Detta kan du kontrollera med kommandot FRE(X) som talar om hur mycket ledigt minne du har.

Detta var lite om Amigans basic. Har du problem eller frågor så skriv en rad till mig.

Matteprogram på ”multipel choice”

Den här veckan skall jag visa ett matematikprogram, som bygger på ”multipel choice”-tekniken. Med ”multipel choice” menas att man får en fråga och till exempel fem alternativ.



Av Christian Lindeberg

Nu skall man visa vilket alternativ som är rätt. Jag skall visa med ett exempel.

Vad är 234+654

A 679
B 763
C 888
D 963
E 865

Nu skall man svara C, vilket är det rätta alternativet.

Jag har gjort programmet bara för addition, eftersom programmet blev för långt med flera räknesätt. Resten av programmet kommer du att få nästa vecka.

Fråga från läsare

Innan jag börjar med programmet skall jag ta en fråga jag fick från Arne i Arlöv. Arne hade skickat ett brev och en diskett låg i brevet. Det var två program, som Arne inte hade lyckas få att fungera.

Jag kontrollerade programmen och kunde inte hitta några fel. Efter provkörning förstod jag att det måste vara något fel. Jag hade två datorer igång. En PC med GWbasic och C-128 med Arnes program. Programmen var lika men gav olika resultat. Var fanns felet? Jo, när jag studerade variablerna kom jag ihåg att det är en stor skillnad mellan GWbasic och VIC-datorernas basic. Om jag i GWbasic skriver följande variabler VMG1, VMG2 och VM, så har jag tre olika variabler. I den basicen, som VIC-datorerna har blir det bara en variabel. Förklaringen är att den basicen bara läser de två första tecknen och struntar i resten. När jag hade ändrat variablerna så fungerade programmet och jag skickade tillbaka disketterna till Arne.

Bara addition

Nu över till programmet. Detta fungerar alltså bara till addition. Resten kommer nästa vecka.

De första 30 raderna är PRINT-satser för att förklara vad som skall göras.

Sedan kommer ett uthopp till en subrutin, som startar i rad 440. Subrutinen kontrollerar först vilken svårighetsgrad du har valt och därefter tar den ut de tal som skall adderas.

I rad 510 tar jag ut den plats som det rätta svaret skall ha.

Raderna 540 till 580 är loopen som skriver ut de olika alternativen. Lägg märke till raderna 560 och 570, där jag har använt mig av uttrycket CHR$(64+I). CHR$(64+I). CHR$(64+I)=A, CHR$864+2)=B o s v på detta sätt så behöver jag bara två rader för att skriva ut bokstäverna A, B, C och E.

Rad 610 kontrollerar om svaret är rätt eller fel.

I rad 570 ser du att programmet gör ett uthopp till en subrutin på raden 700. Denna subrutin tar ut de alternativa svaren som är fel och kontrollerar att samma tal inte kommer upp två gånger.

Lycka till med programmet!



```
10 RANDOMIZE(TIMMER)
20 CLS
30 PRINT:PRINT
40 PRINT:PRINT TAB(10) "MENY"
50 PRINT:PRINT TAB(10) "ÖVERSLAGSBERÄKNING"
60 PRINT:PRINT TAB(10) "VÄLJ RÄKNESÄTT"
70 PRINT:PRINT TAB(10) "1. ADDITION"
80 PRINT:PRINT TAB(10) "2. SUBTRAKTION"
90 PRINT:PRINT TAB(10) "3. MULTIPLIKATION"
100 PRINT:PRINT TAB(10) "4. DIVISION"
110 PRINT:PRINT TAB(10) "VAD VÄLJER DU";
120 INPUT Q
130 CLS
140 PRINT:PRINT
150 PRINT:PRINT TAB(10) "VILKEN SVÅRIGHETSGRAD"
160 PRINT:PRINT TAB(10) "VILL DU HA"
170 PRINT:PRINT TAB(10) "LÄTT=L"
180 PRINT:PRINT TAB(10) "MEDEL=M"
190 PRINT:PRINT TAB(10) "SVÅR=S"
200 PRINT:PRINT TAB(10) "GÖR DITT VAL";
210 INPUT V$
220 CLS
230 PRINT:PRINT
240 PRINT:PRINT TAB(10) "NU KOMMER TIO FRÅGOR"
250 PRINT:PRINT TAB(10) "DU SKALL VÄLJA ETT AV DE"
260 PRINT:PRINT TAB(10) "FEM ALTERNATIVEN"
270 PRINT:PRINT TAB(10) "A, B, C, D ELLER E"
280 PRINT:PRINT TAB(10) "LYCKA TILL!!"
290 PRINT:PRINT TAB(10) "TRYCK PÅ RETURN";
300 INPUT W$
310 IF Q=1 THEN GOSUB 440
320 IF Q=2 THEN GOSUB 670
330 IF Q=3 THEN GOSUB 680
340 IF Q=4 THEN GOSUB 690
350 CLS
360 PRINT:PRINT
370 PRINT:PRINT
380 PRINT:PRINT
390 PRINT:PRINT TAB(10) "DU HADE";R"RÄTT AV 10"
400 PRINT:PRINT TAB(10) "TRYCK PÅ RETURN";
410 INPUT W$
420 CLS
430 GOTO 30
440 CLS
450 IF V$="L" THEN N=100
460 IF V$="M" THEN N=500
470 IF V$="S" THEN N=1000
480 P=INT(RND*N)
490 L=INT(RND*N)
500 SVAR=P+L
510 K=INT(RND*5)+1
520 PRINT:PRINT TAB(10) "VAD ÄR";P;"+";L;
530 T=T+1:PRINT TAB(30) "FRÅGA NR";T
540 FOR I=1 TO 5
550 IF I<>K THEN 570
560 PRINT:PRINT TAB(10) CHR$(64+I);TAB(20);
    SVAR:X(I)=SVAR:GOTO 580
570 PRINT:PRINT TAB(10) CHR$(64+I);TAB(20);:GOSUB 700
580 NEXT I
590 PRINT:PRINT TAB(10) "VILKET ÄR RÄTT ALTERNATIV";
600 INPUT W$
610 IF W$=CHR$(64+K) THEN PRINT:PRINT TAB(10)
    "RÄTT":R=R+1:GOTO 630
620 PRINT:PRINT TAB(10) "FEL"
630 PRINT:PRINT TAB(10) "TRYCK PÅ RETURN";
640 INPUT W$
650 IF T=10 THEN RETURN
660 GOTO 440
670 REM
680 REM
690 REM
700 S=SVAR+INT(RND*SVAR)
710 X(I)=S
720 FOR J=1 TO 5
730 IF J<>I AND X(J)=S THEN GOTO 700
740 NEXT J
750 PRINT S
760 RETURN
```

Skolprogram för glosinlärning

Efterfrågan på skolprogram har ökat. Denna gång fick jag ett önskemål om glosträningsprogram, som sparade glosorna och kunde användas till flera språk.

För att lösa ett sådant problem måste man ta sekventiella filer till hjälp eftersom en lösning med DATA-satser blir stort och onödigt komplicerat.

Programmet har en inläsningsrutin i raderna 170 till 220 och en inskrivningsrutin i raderna 970 till 1040. Programmet är skrivet i GWBasic, så skall du använda en VIC-64 eller C-128. så får du ändra rad 170 och rad 970 på följande sätt om du har diskettstation:

170 OPEN 1,8,1,NAMN$+",S,R"
970 OPEN 1,8,1,NAMN$+",S,W"

Om du har bandspelare så får du skriva på följande sätt,

170 OPEN 1,1,0,NAMN$
970 OPEN 1,1,1,NAMN$

Glöm inte att CLS (tömma skärmen) heter PRINT CHR$(147) på VIC-64 och C-128.

Du kan använda programmet till hur många språk som helst. Tänk bara på att du håller reda på namnen till de filer där du lägger dina glosor. Du kan hela tiden öka på antalet glosor i din fil. Programmet innehåller inga konstigheter utan det är bara att skriva rakt upp och ned.

I programmet upptäcker du att det står till exempel F=1 och P=1. I rad 10 har jag nollställt variablerna. F och P använder jag som flaggor. Om jag redan har laddat in en fil så är F=1 och jag behöver inte läsa in den till minnet igen. P använder jag när jag skall läsa in nya glosor och glosfilen inte finns i minnet. Då läser programmet för st in glosorna för att veta hur många som finns. Därefter så kan man fortsätta med nya glosor. Lycka till med programmet och har du några problem så skriv.




```
10 P=0:F=0:DIM SV$(500),EN$(500):RANDOMIZE(TIMER)
20 CLS
30 ANTAL=0:Q=0:R=0
40 PRINT:PRINT TAB(10)"GLOSTRÄNINGSPROGRAM"
50 PRINT:PRINT TAB(10)"1. TRÄNA GLOSOR"
60 PRINT:PRINT TAB(10)"2. SKRIVA IN NYA GLOSOR"
70 PRINT:PRINT TAB(10)"3. AVSLUTA"
80 PRINT:PRINT TAB(10)"VAD öNSKAS";
90 INPUT Q
100 IF Q<1 AND Q>3 THEN 20
110 ON Q GOTO 120,720,1050
120 CLS
130 IF F=1 THEN 250
140 PRINT:PRINT TAB(10)"VAD HETER FILEN MED DINA GLOSOR"
150 PRINT:PRINT TAB(10)"SKRIV NAMNET OCH TRYCK PÅ RETURN";
160 INPUT NAMN$
170 OPEN"I",#1,NAMN$
180 INPUT#1,N
190 FOR K=1 TO N
200 INPUT#1,SV$(K),EN$(K)
210 NEXT K
220 CLOSE 1
230 F=1
240 IF P=1 THEN 780
250 CLS
260 PRINT:PRINT TAB(10)"GLOSTRÄNING"
270 L=INT(RND*N)+1
280 PRINT:PRINT TAB(10)"TRYCK PÅ S OM DU VILL HA DET SVENSKA ORDET"
290 PRINT:PRINT TAB(10)"ANNARS TRYCK PÅ T";
300 INPUT QQ$
310 IF QQ$="S" THEN 520
320 L=INT(RND*N)+1
330 FEL=0
340 ANTAL=ANTAL+1
350 PRINT:PRINT TAB(10)"VAD BETYDER ";EN$(L);" ";
360 INPUT SVAR$
370 IF SVAR$=SV$(L) THEN R=R+1:GOTO 450
380 PRINT:PRINT TAB(10)"FEL!!!!!"
390 FEL=FEL+1:IF FEL=3 THEN PRINT:PRINT TAB(10)"SVARET= "SV$(L):GOTO 320
400 IF FEL>0 THEN 350
410 PRINT:PRINT TAB(10)"VILL DU SLUTA SÅ TRYCK PÅ M ANNARS RETURN";
420 INPUT QQ$
430 IF QQ$="M" THEN 20
440 GOTO 320
450 PRINT:PRINT TAB(10)"RÄTT!!!!!!!!"
460 PRINT:PRINT TAB(10)"DU HAR SVARAT PÅ";ANTAL;
470 PRINT "FRÅGOR OCH HAR HAFT";R;"RÄTT"
480 PRINT:PRINT TAB(10)"VILL DU SLUTA SÅ TRYCK PÅ M ANNARS RETURN";
490 INPUT QQ$
500 IF QQ$="M" THEN 20
510 GOTO 320
520 L=INT(RND*N)+1
530 FEL=0
540 ANTAL=ANTAL+1
550 PRINT:PRINT TAB(10)"VAD BETYDER ";SV$(L);" ";
560 INPUT SVAR$
570 IF SVAR$=EN$(L) THEN R=R+1:GOTO 650
580 PRINT:PRINT TAB(10)"FEL!!!!!"
590 FEL=FEL+1:IF FEL=3 THEN PRINT:PRINT TAB(10)"SVARET= "EN$(L):GOTO 520
600 IF FEL>0 THEN 550
610 PRINT:PRINT TAB(10)"VILL DU SLUTA SÅ TRYCK PÅ M ANNARS RETURN";
620 INPUT QQ$
630 IF QQ$="M" THEN 20
640 GOTO 520
650 PRINT:PRINT TAB(10)"RÄTT!!!!!!!!"
660 PRINT:PRINT TAB(10)"DU HAR SVARAT PÅ";ANTAL;
670 PRINT "FRÅGOR OCH HAR HAFT";R;"RÄTT"
680 PRINT:PRINT TAB(10)"VILL DU SLUTA SÅ TRYCK PÅ M ANNARS RETURN";
690 INPUT QQ$
700 IF QQ$="M" THEN 20
710 GOTO 520
720 CLS
730 PRINT:PRINT TAB(10)"ÄR DET EN NY FIL";
740 INPUT QQ$
750 IF QQ$="J" THEN GOSUB 920
760 IF F=1 THEN 780
770 P=1:GOTO 120
780 CLS
790 PRINT:PRINT TAB(10)"INSKRIVNING AV NYA GLOSOR"
800 PRINT:PRINT TAB(10)"SKRIV FöRST GLOSAN PÅ SVENSKA"
810 PRINT:PRINT TAB(10)"SEDAN PÅ DET FRÄMMANDE SPRÅKET"
820 N=N+1
830 PRINT:PRINT TAB(10)"DET SVENSKA ORDET";
840 INPUT SV$(N)
850 PRINT:PRINT TAB(10)"DET FRÄMMANDE ORDET";
860 INPUT EN$(N)
870 PRINT:PRINT TAB(10)"FLER GLOSOR (J/N)";
880 INPUT QQ$
890 IF QQ$="J" THEN 820
900 GOSUB 920
910 GOTO 20
920 CLS
930 F=1
940 PRINT:PRINT TAB(10)"VAD HETER FILEN MED DINA GLOSOR"
950 PRINT:PRINT TAB(10)"SKRIV NAMNET OCH TRYCK PÅ RETURN";
960 INPUT NAMN$
970 OPEN"O",#1,NAMN$
980 PRINT#1,N
990 FOR K=1 TO N
1000 PRINT#1,SV$(K)
1010 PRINT#1,EN$(K)
1020 NEXT K
1030 CLOSE 1
1040 RETURN
1050 END
```

Program för lösning av enklare ekvationer

Den här veckan fortsätter matematiken med något enklare. Sven A, Malmö vill ha ett program, som ger lösningar på enkla ekvationer av typen Y=K*X+L. Jag har skrivit ihop ett program, som skall klara de flesta ekvationer.



Hem DATORER

Av Christian Lindeberg

Det kom också ett brev från Enar N i Markaryd, som hade ett speciellt problem. Enar har en ABC 802 och undrar hur han skall ändra i ett program. Programmet det gäller är från maj 1987 och handlar om att hantera sin kassabok med hjälp av dator. Problemet är att programmet lagrar uppgifterna på disk och hur skall detta göras på en ABC 802?

I programmet ser det ut så här:

580 OPEN "I",#1,"KASSABOK"
●
●
●
670 CLOSE 1
●
690 OPEN "O",#1, "KASSABOK"
●
●
780 CLOSE 1

Om jag kommer ihåg riktigt, det är fem år sedan jag satt vid en ABC-dator, så skall man skriva på följande sätt:

580 OPEN "KASSABOK" ASFILE 1
●
●
670 CLOSE 1
●
690 PREPARE "KASSABOK" ASFILE 1
●
●
780 CLOSE 1

När du lagrar programmet så skriver du SAVE KASSA. Om du vill ha filen på en annan drive så skriver du SAVE SF1: KASSA. Jag hoppas att det är rätt. Om några av mina läsare har en annan uppfattning så skriv en rad till mig.

I Enars brev fanns en fråga som jag inte förstod. Du frågar om programmet är lämpligt för strängvariabler. Kan du utveckla det lite mer i ett brev så kanske jag kan besvara frågan.

Letar fram roten

Nu över till ekvationerna. Jag har gjort ett program, som letar fram den rätta roten till ekvationen. Principen är att datorn kontrollerar när ekvationen blir noll och hoppar då till en utskriftsdel. Om vi tar ekvationen Y=2*X-4, då letar datorn tills den hittar lösningen till 2*X-4=0, vilket i det är fallet blir X=2. Programmet går i tre steg. Först grovsorterar programmet ut var någonstans mellan -100 och 100 roten ligger. Är det en jämn rot så hoppar den direkt till utskrift, annars tar den intervallet där X-värdet ligger och testar i steg om 0,1. Detta kan du se i rad 280. Sedan upprepas samma sak och i nästa led testas X i steg om 0,01.

Lägg märke till rad 90 där jag har använt funktionen DEF FNA(X). Denna är mycket användbar när man löser ekvationer.

Två andra funktioner, som jag också använt är SGN och ABS. SGN() blir -1 om talet är negativt, 0 om det är 0 och 1 om det är ett postivt tal.

ABS() gör alla tal positiva.

Har du några frågor på programmet så skriv en rad.

Lycka till!

```
10 DIM T(200)
20 CLS:N=0:F=0
30 PRINT :PRINT TAB(10);"LöSNING AV ENKEL EKVATION"
40 PRINT :PRINT TAB(10);"AV TYPEN Y=K*X+L"
50 PRINT :PRINT TAB(10);"VILKET VÄRDE HAR K";
60 INPUT K
70 PRINT :PRINT TAB(10);"VILKET VÄRDE HAR L";
80 INPUT L
90 DEF FNA(X)=K*X+L
100 FOR I=-100 TO 100
110 N=N+1
120 T(N)=FNA(I)
130 IF F=0 THEN 160
140 IF FNA(I)=0 THEN SVAR=I:GOTO 430
150 IF SGN(T(N))<>MIN THEN E(1)=I-1:E(2)=I+1:GOTO 190
160 F=1
170 MIN=SGN(T(N))
180 NEXT I
190 N=0
200 FOR I=E(1) TO E(2) STEP .1
210 N=N+1
220 T(N)=FNA(I)
230 IF FNA(I)=0 THEN SVAR=I:GOTO 430
240 IF SGN(T(N))<>MIN THEN E(1)=I-.1:E(2)=I+.1:GOTO 270
250 MIN=SGN(T(N))
260 NEXT I
270 N=0
280 FOR I=E(1) TO E(2) STEP .01
290 N=N+1
300 T(N)=FNA(I)
310 IF ABS(FNA(I))<.00001 THEN SVAR=I:GOTO 430
320 IF SGN(T(N))<>MIN THEN E(1)=I-.01:E(2)=I+.01:GOTO 360
330 MIN=SGN(T(N))
340 NEXT I
350 REM
360 N=0
370 FOR I=E(1) TO E(2) STEP .001
380 N=N+1
390 T(N)=FNA(I)
400 IF ABS(T(N))<.001 THEN SVAR=I:GOTO 430
410 MIN=SGN(T(N))
420 NEXT I
430 REM
440 PRINT :PRINT TAB(10);"I EKVATIONEN";K;"* X";
450 IF SGN(L)=1 THEN PRINT "+";
460 PRINT L;"=0"
470 PRINT :PRINT TAB(10);"ÄR X=";INT(SVAR*1000+.5)/1000
480 END
```

I dag avslutar vi mattekursen

Snart är sommaren här och då brukar jag göra ett sommaruppehåll, men inte denna sommar. Jag tänker att under sommarveckorna köra en kurs i basicprogrammering rån början.



Av Christian Lindeberg

Alla ni som känner att ni har några luckor i ert kunnande får nu chansen att fylla igen dessa.

Den här veckan avslutar jag matematikprogrammen, med att ta fram ett program för att lösa andragradsekvationer och för att lösa ekvationssystem.

Programdelen för andragradsekvationer löser ekvationer av typen:
$A*X^2+B*X+C=0$
Ekvationssystemdelen löser följande ekvationssystem:
$A*X+b*Y=C$
$D*X+E*Y=F$
Andragradsekvationerna löses efter den allmänna metoden vars formel ser ut på följande vis:
$X1=-B/(A*2)+SQR((B/(A*2))^2-C/A))$
$X2=-B/(A*2)-SQR((B/(A*2))^2-C/A))$
Båda programdelarna kontrollerar också att systemen har en lösning.

Liten läxa

Nästa vecka skall jag börja med basickursen och jag tänker att försöka ge er lite läxa under veckan.

Den första läxan blir att funder ut vilka radnummer som skall finnas där jag har skrivit XXX.

Här kommer läxan:

```
10 GOTO XXX
20 PRINT "ÄR";
30 GOTO XXX
40 PRINT "!"
50 GOTO XXX
60 PRINT "DATA";
70 GOTO XXX
80 PRINT "KUL";
90 GOTO XXX
100 END
```

Din dator skall nu skriva ut följande på skärmen:
DATA ÄR KUL!
Lycka till med läxan!

```
10 CLS
20 PRINT :PRINT TAB(10);"MENY"
30 PRINT :PRINT TAB(10);"1. ANDRAGRADSEKVATIONER"
40 PRINT :PRINT TAB(10);"2. EKVATIONSSYSTEM MED TVA OBEKANTA"
50 PRINT :PRINT TAB(10);"3. AVSLUTA"
60 PRINT :PRINT TAB(10);"VAD öNSKAS";
70 INPUT Q
80 IF Q<1 AND Q>3 THEN 10
90 ON Q GOTO 110,500,820
100 GOTO 10
110 CLS
120 PRINT :PRINT TAB(10);"ANDRAGRADSEKVATIONER"
130 PRINT :PRINT TAB(10);"                    2";
140 PRINT TAB(10);"FORMEN SKALL VARA: A*X+B*X+C=0"
150 PRINT :PRINT TAB(10);"A=";
160 INPUT A
170 PRINT :PRINT TAB(10);"B=";
180 INPUT B
190 PRINT :PRINT TAB(10);"C=";
200 INPUT C
210 PRINT :PRINT
220 IF A=0 THEN 320
230 B=B/A
240 C=C/A
250 A=1
260 IF B*B-4*C<0 THEN 460
270 PRINT :PRINT TAB(10);"X1=";0-B/2+(SQR(B*B-4*C))/2
280 PRINT :PRINT TAB(10);"X2=";0-B/2-(SQR(B*B-4*C))/2
290 PRINT :PRINT TAB(10);"TRYCK PA RETURN";
300 INPUT QQ$
310 GOTO 10
320 IF B=0 THEN GOTO 370
330 PRINT :PRINT TAB(10);"X=";0-C/B
340 PRINT :PRINT TAB(10);"TRYCK PA RETURN";
350 INPUT QQ$
360 GOTO 10
370 IF C=0 THEN GOTO 420
380 PRINT :PRINT TAB(10);"EKVATIONEN SAKNAR LöSNING"
390 PRINT :PRINT TAB(10);"TRYCK PA RETURN";
400 INPUT QQ$
410 GOTO 10
420 PRINT :PRINT TAB(10);"EKVATIONEN SATISFIERAS AV ALLA X-VÄRDEN"
430 PRINT :PRINT TAB(10);"TRYCK PA RETURN";
440 INPUT QQ$
450 GOTO 10
460 PRINT :PRINT TAB(10);"TRYCK PA RETURN";
470 INPUT QQ$
480 GOTO 10
490 END
500 CLS
510 PRINT :PRINT TAB(10);"EKVATIONSSYTEM MED TVA OBEKANTA"
520 PRINT :PRINT TAB(10);"FORMEN SKALL VARA"
530 PRINT :PRINT TAB(10);"A*X+B*Y=C"
540 PRINT :PRINT TAB(10);"D*X+E*Y=F"
550 PRINT :PRINT TAB(10);"A=";
560 INPUT A
570 PRINT :PRINT TAB(10);"B=";
580 INPUT B
590 PRINT :PRINT TAB(10);"C=";
600 INPUT C
610 PRINT :PRINT TAB(10);"D=";
620 INPUT D
630 PRINT :PRINT TAB(10);"E=";
640 INPUT E
650 PRINT :PRINT TAB(10);"F=";
660 INPUT F
670 IF B*D-A*E=0 THEN 730
680 PRINT :PRINT TAB(10);"X=";(B*F-C*E)/(B*D-A*E)
690 PRINT :PRINT TAB(10);"Y=";(C*D-A*F)/(B*D-A*E)
700 PRINT :PRINT TAB(10);"TRYCK PA RETURN";
710 INPUT QQ$
720 GOTO 10
730 IF B*F-E*C=0 THEN 780
740 PRINT :PRINT TAB(10);"EKVATIONEN SAKNAR LöSNING"
750 PRINT :PRINT TAB(10);"TRYCK PA RETURN";
760 INPUT QQ$
770 GOTO 10
780 PRINT :PRINT TAB(10);"EKVATIONEN SAKNAR ENTYDIG LöSNING"
790 PRINT :PRINT TAB(10);"TRYCK PA RETURN";
800 INPUT QQ$
```

Basicskola i sommar



Av Christian Lindeberg

Nu har juni månad börjat och även sommaren. Hela sommaren kommer jag att ha en basicskola. Jag skall försöka lära er en del basic och några olika tips, som kan vara bra att ha när man programmerar.

Först skall jag svara på ett brev från Leif Ax i Staffanstorp. Leifs fråga är lätt att besvara, men komplicerad att göra ett program till.

Leifs fråga lyder så här:

Kan jag ta en fil från mitt ordbehandlingsprogram och tillsammans med gloträningsprogrammet från den 22 april göra rättningar i det, som jag skrivit med hjälp av ordbehandlingsprogrammet. Svaret är givetvis ja, MEN programmet, som skall göra jobbet blir mycket komplicerat och ligger långt utanför den här artikeln.

Tecken för tecken

Principen är att du gör ett program, som läser in din text tecken för tecken och delar upp den i ord. Detta gör den genom att känna av mellanslaget mellan orden. Mellanslagen har ASCII-koden 32 decimalt. När man sedan lagrat orden i indexerade variabler kan man börja att leta upp och jämföra orden. De riktiga orden bör man då ha lagrade i en egen fil. Denna fil bör vara randomfil för att inte ta upp för mycket minne. Sekventiella filer tar upp mycket minne eftersom hela filen måste laddas in på en gång.

I någon av mina tidigare artiklar finns ett program, som visar hur man känner av de olika orden. Jag skall leta efter artikeln och återkomma under sommaren.

Nu över till basicskolan. Du fick en läxa förra veckan. Jag hoppas att du har hunnit lösa uppgiften. Här kommer läxan igen.

```
10 GOTO XXX
20 PRINT "ÄR";
30 GOTO XXX
30 PRINT "!"
50 GOTO XXX
60 PRINT "DATA";
70 GOTO XXX
80 PRINT "KUL";
90 GOTO XXX
100 END
```

Du skulle sätta in riktiga radnummer istället för XXX, så att utskriften blev DATA ÄR KUL! Vad man då måste tänka på är att GOTO-instruktionen är ett ovillkorligt hopp, vilket ofta man glömmer bort. GOTO gör att datorn hoppar till ett visst radnummer och FORTSÄTTER därifrån vidare mot högre radnummer.

Det första GOTO i rad 10 skall alltså hoppa till PRINT-satsen, som skriver ut ordet DATA. Detta sker i rad 60, då skall rad 10 se ut så här:

```
10 GOTO 60
```

När datorn kommer till rad 60 så skriver den ut ordet DATA och OCH GÅR DÄREFTER TILL RAD 70. I rad 70 skall den sedan få instruktion att hoppa till nästa rad med orden ÄR. Ordet ÄR befinner sig i rad 20, så att rad 70 kommer att se ut så här:

```
70 GOTO 20
```

Från rad 20 går den sedan till rad 30 och får där instruktionen att hoppa till rad 80. Från rad 80 går programmet till rad 90 och härifrån skall den hoppa till rad 40. Från rad 40 går programmet till rad 50 där den skall hoppa till END på rad 100. Så komplett ser programmet ut så här:

```
10 GOTO 60
20 PRINT "ÄR";
30 GOTO 80 40 PRINT"!"
50 GOTO 100
60 PRINT "DATA";
70 GOTO 20
80 PRINT "KUL";
90 GOTO 40
100 END
```

Kör nu programmet och se att det fungerar.

Det finns många olika programmeringsspråk och de har olika sätt att skrivas. En del språk behöver radnummer men det stora flertalet klarar sig bra utan.

Använd radnummer!

Basic är ett språk, som i de flesta fall måste ha radnummer. Jag säger i de flesta, eftersom AMIGA-basic till exempel inte behöver några radnummer. Oavsett vilken basic du använder så skulle jag föreslå att du som nybörjare alltid använder radnummer.

I programmet har jag använt två instruktioner GOTO och PRINT. GOTO har jag förklarat, men inte PRINT. PRINT använder jag i exemplet för att skriva ut de olika orden. Lägg märke till att när jag vill att datorn skall skriva ut ord så måste jag ha citationstecken på båda sidor om ordet.

Räkneinstruktion

PRINT kan också användas som en räkneinstruktion och bokstäver utan citationstecken blir variabler. Låt oss titta på en del exempel.

Skriv ner mina exempel på din dator och kontrollera själv.

PRINT 3★2 tryck på RETURN.

Nu kommer datorn att skriva ut talet 6. PRINT har alltså gett datorn i uppgift att räkna ut vad 2★3 är och skriva ut det på skärmen. Datorn kan hantera alla räknesätt, så skriver du följande:

PRINT 2+196/7 så svarar datorn med 30. Datorn gör först divisionen och sedan additionen. Nästa vecka skall jag fortsätta med PRINT och variabler. Lycka till!

Något om DOS till PC-datorer



Hem DATORER

Av Christian Lindeberg

Basicskolan fortsätter och jag tänker också fortsätta med PC-datorer genom att skriva om DOS-systemen. Det underlättar mycket om man kan arbeta i DOS-miljö på en PC-dator.

Först ska jag ta läxan från förra gången. Ni fick två rader, som såg ut så här:

```
10 A$="236 00 HÖLLVIKEN"
20 B$="231 00 TRELLEBORG"
```

Sedan skulle ni fortsätta programmet så att datorn adderade ihop talen 236 och talet 231. Här kommer svaret och även på den svårare delen med 23600+23100.

```
10 A$="236 00 HÖLLVIKEN"
20 B$="231 00 TRELLEBORG"
30 C$=MID$(A$,1,3):REM 236
40 D$=MID$(B$,1,3):REM 231
50 E$=MID$(A$,5,2):REM 00
60 F$=C$+E$ :REM 23600
70 G$=D$+E$ :REM 23100
80 K=VAL(C$) :REM ÖVERFÖR TILL NUMERISK VARIABEL
90 L=VAL(D$) :REM ÖVERFÖR TILL NUMERISK VARIABEL
100 M=VAL(F$):REM ÖVERFÖR TILL NUMERISK VARIABEL
110 N=VAL(G$) :REM ÖVERFÖR TILL NUMERISK VARIABEL
120 PRINT K+L
130 PRINT M+N
140 END
```

Jag har lagt REM-satser (REMARK) efter raderna för att du skall se vad jag plockar ut av strängen. I raderna 120 och 130 lägger jag bara ihop talen. Svaren skall bli 467 och 46700.

Stränguttryck

Det finns ett stränguttryck till, som jag tycker man ska kunna, och det är LEN(). LEM() räknar ut hur många tecken det finns i strängen. Vi tar följande sträng:

```
10 A$="SOMMAR"
20 PRINT LEN(A$)
```

Nu kommer datorn att svara med talet 6, som talar om att det finns 6 tecken i strängen A$. Ta och pröva med några andra strängar.

I dag ska vi ta en till instruktion, nämligen IF...THEN, som testar om något är sant eller falskt. Är det falskt så hoppar den vidare, men är det sant går den till uttrycket efter THEN. Ett exempel:

```
10 INPUT A
20 IF A=6 THEN 40
30 GOTO 10
40 PRINT "TALET ÄR SEX"
```

Om du i raden 10 ger värdet 6, så hoppar datorn till rad 40 annars går den till rad 30 där den hoppar till rad 10. Om du har en basic med IF..THEN..ELSE så kan du skriva programmet så här:

```
10 INPUT A
20 IF A=6 THEN PRINT"TALET ÄR SEX" ELSE 10
```

Testa "HEJ"

Läxan till nästa vecka blir att skriva ovanstående program men använda strängar och testa ordet "HEJ".

Nu över till DOS-system till PC-datorer. Jag kommer i fortsättningen att kalla den för DOS-skolan.

När datorn startar måste det finnas DOS på disketten eller hårddisken DOS är operativsystemet, som ofta kallas MS-DOS efter tillverkaren MicroSoft, vi ska lära oss att behärska.

Datorn har startat och på skärmen ser du antingen: A> eller A: (diskettsystem med en eller två diskettstationer), eller: C> eller C: (Hårddisk).

Jag kommer bara att beröra kommandon, som finns i versioner över 2.00. Detta kan du testa genom att skriva VER och trycka på RETURN. Nu svarar datorn med den DOS-version du har.

Formatering

Den första åtgärd vi ska ta är att formatera några disketter. Formateringen är till för att anpassa disketterna till ditt DOS-system. Om du har två diskettstationer så sätter du systemdisketten i A: och en oformaterad diskett i B: På en del PC-datorer kallas A: för 0: och B: för 1:.

Nu skriver du följande: FORMAT B: /S. Om du har hårddisk så skriver du så här: FORMAT A: /S.

Detta kommando består av tre delar. FORMAT, som är själva kommandot och det andra som är parametern, som talar om vilken diskettstation den oformaterade disketten sitter i. Det tredje uttrycket är en parameter, som talar om att du vill att operativsystemet DOS också skall läggas på disketten.

Om du inte vill ha DOS på disketten, hoppar du över /S. Kom ihåg att en formatering tar bort all information som har funnits på disketten.

Kopiera

När formateringen är klar så talar datorn om detta på följande sätt:

```
Formating... formating complete
System transferred
362496 bytes total disk space
40960 bytes used by system
321536 bytes available on disk
Format another (Y/N)
```

Om du vill formatera fler så svarar du med Y annars med N. När du har formaterat dina skivor är det tid för att kopiera och detta finns det en del kommando för. Nästa gång skall jag gå igenom DIR, COPY, DISKCOPY och CHKDSK. Lycka till!

Simulering med katter och möss

Simuleringsprogram skall det handla om den här veckan. Simulering sätter de flesta i samband med att vara "låtsassjuk", men i datorsammanhang är det en analysmetod där man studerar en modell av den verkliga händelsen.



HEMDATORER

Innan jag börjar med programmet skall jag svara på en fråga från signaturen KAP i Lund. KAP har en Amstrad PCWB256 med mikroprocessor Z80A. Z80 är fadern till ABC-80, TRS-80, Sinclairs datorer med Spectrum o s v. Problemet, som KAP har är att få RND-funktionen att fungera. I ditt brev fanns en lista på det program som du försökte att köra. I rad 10 har du skrivit DIM AB(35). Det skall vara DIM A(35). I rad 20 skall du använda funktionen för att rensa skärmen. CHR$(147) gäller för en VIC-64, försök med CLS eller titta i din instruktionsbok. RND-funktionen kan skrivas på två sätt i basic. Antingen RND eller RND(0) kontrollera i din instruktionsbok hur din dator vill ha det.

Dagens program

I dagens program kommer jag att använda RND-funktionen. Denna funktion använder man ofta när man simulerar försök.

Dagens problem är följande:

Tänk er att det finns sju stycken burar på rad. De är sammanbundna med en liten gång så att en mus kan gå från bur till bur. I bur nummer 1 och 7 sitter en katt, som älskar att äta möss. Nu kommer problemet. Musen börjar i bur 3 och flyttar sig varje minut antingen ett steg åt höger eller ett steg åt vänster. Hur lång tid tar det innan musen är uppäten?

Stanna här ta fram papper och penna och försök räkna ut svaret. Hur går det? Har du fått fram ett svar?

Simulering

Troligtvis inte. Hur löser man nu ett sådant här problem om man inte kan matematiken bakom problemet. Jo, man använder sig av simulering. Vi använder ett stort antal möss, ju fler desto säkrare resultat. Jag skriver programmet i GWbassic, som är den basic-dialekt, som finns på PC-datorer.

Burarna får variabeln X, tiden variabel T, antalet variabeln N och förflyttningen variabeln S. I början skall X ha värdet 3, därför att det är startburen för musen. Det enda problem som dyker upp är hur skall vi få datorn att flytta musen ett steg åt höger eller vänster.

Om musen börjar i bur 3 så skall vi antingen minska eller lägga till 1, så att musen hamnar i bur 2 eller bur 4. Vi skall alltså se till att datorn tar fram antingen +1 eller −1. Vi konstruerar en funktion med hjälp av RND.

S=INT(RND★2) ger antingen 0 eller 1. Om vi nu lägger till ★2, så här:

S=INT(RND★2)★2, kommer vi att få antingen 2 eller 0, så nu har vi bara att minska med 1 så är vi framme vid målet.

S=INT(RND★2)★2−1.

I variabeln S får vi nu antingen +1 eller −1. Nu är vi klara att skriva ner hela programmet.

```
10 RANDOMIZE(TIMER)
20 CLS
30 X=3
40 N=N+1
50 T=T+1
60 S=INT (RND★2)★2−1
70 X=X+S
80 IF X=1 THEN 30
90 IF X=7 THEN 30
100 PRINT "ÖVERLEVNADS-
TID"; T/N; "ANTALET MÖSS";N
110 GOTO 50
```

I rad 10 har jag en rad som bara behövs i GWbasic, för att få ordentliga slumptalsserier. Rad 20 tömmer skärmen och i raderna 80 och 90 testar jag om musen har kommit till bur 1 eller 7 för att i så fall är den död och vi stoppar i en ny mus i bur nummer 3.

Åtta minuter

Lycka till med musdödandet. Det matematiskt korrekta svaret är 8 minuter. En gång när jag körde 500.000 möss fick jag svaret 7,99 som är ett bra svar. Nästa gång tänkte jag vi skulle simulera väntetider hos en ensam herrfrisör, som tar 20 minuter för att klippa en kund. Hur lång blir längsta kön? Glöm inte att han har inte kunder hela tiden. Bra jogging för hjärnan.

DOS-skola

Nu över till DOS-skolan, som lider mot sitt slut. Detta är näst sista gången för DOS-skolan, men jag kommer naturligtvis även i fortsättningen att besvara frågor och ta upp nya saker, som jag stöter på.

Den här gången skall det handla om DOS-kommandon och jag skall göra en uppräkning av de flesta denna och nästa gång. Jag delar upp dem i tre grupper. Grupp ett är de kommandon, som fanns med från början. I grupp två kommer kommandona från DOS-2 och i tredje gruppen finns kommando från DOS-3. Har du till exempel DOS 3.00 eller högre så har du alla kommandon.

COPY — kopierar filer på hårddisk och disketter
DATE — visar och ställer datum i klockan
DEL — raderar filer
DIR — visar innehållet på skivorna
ERASE — raderar filer
PAUSE — används för att stoppa och därefter visa meddelande
REM — visar meddelande utan att påverka kommandofilen
REN — döper om filer och ändrar namn
TIME — ställer klockan och visar tiden
TYPE — visar innehållet av en fil om den är skriven i ASCII-kod

Med DOS−2 tillkom följande kommando:
ASSIGN ansluter viss skivenhet

Om du har en applikation på din hårddisk C:, som fordar att du har en programdiskett i A: och en datadiskett i B: så kan du ge följande kommando innan du startar ditt program:
ASSIGN A=C B=C

Alla referenser till A: och B: kommer att gå till C:.
BREAK styr hur DOS kontrollerar avbrott, CTRL och C
CHDIR (CD) byter aktuell katalog på skivan
CLS tömmer skärmen (samma som i GWbasic)
CTTY sätter DOS till fjärrstyrning genom att styra om terminalen, använd inte detta om du inte vet vad du gör
ECHO kopplar av och på kommandoutskrift samt visar meddelanden
FOR används i avancerade kommandofiler.

Nästa gång fortsätter jag med fler kommandon.

Frisörens problem löses via datorn

Denna vecka har turen kommit till frisören och hans problem med sina kunder. Förutsättningarna är följande:
Salongen har öppet mellan klockan 09.00 och 18.00 (frisören äter inte lunch han bantar).
Under denna tid kommer det max 27 kunder, fler hinner han inte. Han kan bara klippa en i taget.
En klippning tar 20 minuter.
Alla kunder, som kommer innan 18.00 blir klippta.
Kunderna klipps i tur och ordning.



HEMDATORER

Vi vill ha svar på följande frågor.
Hur länge får kunden vänta i genomsnitt?
Hur lång blir längsta väntetiden?
Hur lång blir längsta kön?
Vi börjar med att dimensionera programmet för max 27 kunder i rad 20.
I raderna 50 till 70 tar datorn fram ankomsttiden för det antal kunder du har uppgett i rad 30. I variabeln A(1) till A(N) kommer det nu att ligga ankomsttiden för kunderna. Om det i variabeln A(10) ligger talet 182 betyder det att han kom 182 minuter efter klockan 09.00 vilket blir klockan 12.02 o s v.
I nästa del av programmet från rad 80 till rad 170 sorterar vi ankomsttiderna i ordning så att den kund som kommer först ligger i A(1) nästa i A(2) o s v.

Startid

Starttiden för första kunden blir när första kunden har kommit alltså A(1), som blir starttid som läggs i S(1). För kund nummer 6 blir starttiden A(6) plus starttiden för S(5) plus 20 minuter för klippningen alltså S(6)=S(5)+20. Rent allmänt blir formeln S(1)=S(1−1)+20.

Väntetid

Denna loop för starttiderna hittar du i raderna 180 till 240.
I raderna 250 till 300 tar datorn och räknar ut den sammanlagda väntetiden och lägger denna i variabeln T. Varje väntetid får man genom att minska kundens ankomsttid A(1) med starttiden S(1) denna läggs i variabeln V(1), som sedan summeras i variabeln T. I rad 300 delas den totala väntetiden T med antalet kunder N och resultatet skrivs ut.
I rad 320 till rad 360 tar vi fram den längsta väntetiden M. I rad 320 antar vi att i starten är den längsta väntetiden V(1). I loopen jämför man sedan alla värden med varandra för att få fram den längsta väntetiden M.
Rad 370 skriver datorn ut väntetiden.

Hur många i kön?

Nu återstår bara att räkna ut hur många kunder det var i den längsta kön. Detta gör man genom att ta värdet i M och dela med 20, som är klippningstiden och lägga till 1, så får man antalet kunder i den längsta kön. Skrivningen hittar du i rad 390.
Har du själv någon simulering skriv och berätta.
Nu skall jag fortsätta med DOS-skolan. Jag ser att det blir inte sista gången, som jag hotade med tidigare. Basicdelen tog lite mer plats än jag räknade med.

DOS-2

Jag fortsätter där jag slutade förra gången och har då kommit till följande kommando under DOS-2.
GOTO används i avancerade kommandofiler för hopp
IF används för att kontrollera logiska villkor (jämför basic)
MKDIR (MD) skapar ny underkatalog på skivan.
PATH talar om för DOS var den kan hitta program. Om jag vill att datorn skall leta i olika underbibliotek efter program eller data så kan jag använda mig av PATH. Vi kan till exempel säga att i underbiblioteken KALKYL, KALKYL\EL och ORD vill jag att datorn automatiskt ska leta efter filer så skriver jag så här:
PATH\KALKYL;\KALKYL/EL;\ORD
PRINT gör utskrift till skrivare
PWD visar aktuellt katalognamn (finns inte i alla DOS)
RECOVER återställer fil eller bibliotek. Om du har en diskett som innehåller dåliga sektorer kan du skriva RECOVER A: om det är ett program, som till exempel heter SKRIV.ORD och har några dåliga sektorer så skriver du RECOVER SKRIV.ORD och trycker på RETURN.

Hör ihop

RESTORE OCH BACKUP är två kommandon som hör ihop. BACKUP använder man för att ta säkerhetskopia på sitt system eller dela av det. RESTORE är att lägga tillbaka de säkerhetskopierade filerna, som är kopierade med BACKUP.
Vi skall säkerhetskopiera alla filer i underbiblioteket som heter KALKYL\EL. Detta gör jag genom att skriva:
BACKUP C:\KALKYL\EL A:
Se till att ha en TOM formaterad diskett i A: För att sedan läsa tillbaka med RESTORE skriver man så här:
RESTORE A: C:\KALKYL\EL
Nästa gång avslutar jag DOS-skolan.

```
10 CLS:RANDOMIZE(TIMER)
20 DIM A(27),S(27),V(27)
30 INPUT "ANTAL KUNDER PER DAG (MAX 27 ST)";N
40 PRINT:PRINT
50 FOR I=1 TO N
60 A(I)=INT(541*RND)
70 NEXT I
80 FOR J=1 TO N-1
90 M=J
100 FOR K=J+1 TO N
110 IF A(K)>=A(M) THEN 130
120 M=K
130 NEXT K
140 X=A(J)
150 A(J)=A(M)
160 A(M)=X
170 NEXT J
180 S(1)=A(1)
190 FOR I=2 TO N
200 IF S(I-1)+20>=A(I) THEN 230
210 S(I)=A(I)
220 GOTO 240
230 S(I)=S(I-1)+20
240 NEXT I
250 T=0
260 FOR I=1 TO N
270 V(I)=S(I)-A(I)
280 T=T+V(I)
290 NEXT I
300 PRINT "DEN GENOMSNITTLIGA VÄNTETIDEN";INT(T/N+.5)
310 PRINT:PRINT
320 M=V(1)
330 FOR I=2 TO N
340 IF V(I)<=M THEN 360
350 M=V(I)
360 NEXT I
370 PRINT "LÄNGSTA VÄNTETIDEN";M
380 PRINT:PRINT
390 PRINT "DEN LÄNGSTA KÖN BLEV";INT(M/20)+1;"PERSONER"
```

Så här gör du en månadskalender



Christian Lindeberg

HEMDATORER

Jag fick ett brev från Roland N i Viken. Han har köpt en Apple II Europlus. Datorn är av äldre datum men fortfarande en bra dator att lära sig på. Om jag kommer ihåg riktigt så använder det ett operativsystem som heter CP/M 2.0 eller liknande.

Det finns mycket program till detta operativsystem. Om någon annan läsare har samma typ av dator, så förmedlar jag Rolands adress. Roland har ett problem med sina basicprogram. Han skickade med en del listningar.

Första anledningen till att dina program inte fungerar är rad 40. Ta bort den för den gör att datorn stannar där och inte fortsätter programmet.

Rad 230 är fel titta efter i boken vad där skall stå.

Rad 310 tar du bort.

Rad 340 skriver du RND(X) i stället för RND.

Programmet verkar onödigt komplicerat.

Nästa vecka skall jag skriva ett tipsprogram, som tar ut tipsrader och där du kan bestämma sannolikheter själv.

Den här veckan har jag gjort ett program, som ger dig en månadskalender för valfri månad från och med 1582.

SEPTEMBER 1989

MÅN	TIS	ONS	TOR	FRE	LÖR	SÖN
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Ovanstående får du om du skriver in 1989 på frågan om år och talet 9 som svar på vilken månad.

Skottår

Den svårast biten i programmet är bara kontrollen av skottår annars är det en ganska enkel matematik.

Om du har en dator med en skärm som är 40 tecken bred så får du ändra i raderna 400 och 470. I båda raderna står *6 ändra detta till *5. Rad 520 ändrar du till bara PRINT.

Lycka till med kalendern!

```
10 CLS:PRINT :PRINT
20 PRINT "UTSKRIFT AV KALENDER"
30 PRINT :PRINT "VALFRI MANAD OCH AR"
40 FOR I=1 TO 5000:NEXT I
50 DIM M$(12)
60 S$="-------------------------------------------"
70 T$=":     :     :     :     :     :     :     :"
80 FOR I=1 TO 7
90 READ D$(I)
100 NEXT I
110 DATA MANDAG,TISDAG,ONSDAG,TORSDAG,FREDAG,LöRDAG,SöNDAG
120 FOR I=1 TO 12
130 READ M$(I)
140 NEXT I
150 DATA JANUARI,FEBRUARI,MARS,APRIL,MAJ,JUNI
160 DATA JULI,AUGUSTI,SEPTEMBER,OKTOBER,NOVEMBER,DECEMBER
170 CLS:PRINT :PRINT
180 INPUT"ANGE AR, MEN INTE FöRE 1582";AR
190 IF AR<1582 THEN 170
200 INPUT"ANGE MANADENS NUMMER";N
210 MAN$=M$(N)
220 ST=AR
230 IF N<>2 THEN 280
240 IF AR/4<>INT(AR/4) THEN ML=28:GOTO 300
250 ML=29
260 IF AR/100=INT(AR/100) AND AR/400<>INT(AR/400) THEN ML=28
270 GOTO 300
280 IF (N=4 OR N=6 OR N=9 OR N=11) THEN ML=30:GOTO 300
290 ML=31
300 D=0
310 IF N<3 THEN N=N+12:AR=AR-1
320 P=D+2*N+INT(.6*(N+1))+AR+INT(AR/4)-INT(AR/100)+INT(AR/400)+2
330 P=INT((P/7-INT(P/7))*7+.5)
340 IF P=0 THEN P=7
350 DAG=1-P
360 CLS
370 PRINT :PRINT "  ";MAN$;TAB(39);ST
380 PRINT " ";S$
390 FOR KOL=1 TO 7
400 PRINT TAB(3+(KOL-1)*6);LEFT$(D$(KOL),3);
410 NEXT KOL
420 PRINT
430 PRINT TAB(2);S$
440 FOR RAD=1 TO 6
450 PRINT " ";T$
460 FOR KOL=1 TO 7:DAG=DAG+1
470 PRINT TAB(2+(KOL-1)*6);":";
480 IF DAG>ML OR DAG<=0 THEN 510
490 IF DAG<10 THEN PRINT "  ";DAG;
500 IF DAG>9 THEN PRINT " ";DAG;
510 NEXT KOL
520 PRINT TAB(44);":"
530 PRINT " ";S$
540 NEXT RAD
550 PRINT " EN NY MANAD TILL (J/N)";
560 INPUT SVAR$
570 IF SVAR$="J" OR SVAR$="j" THEN 170
580 END
```

Tipsprogram och säkerhetskopiering

Förra veckan lovade jag Roland N i Viken ett tipsprogram. Jag har gjort ett program där du kan låta datorn slumpmässigt ta ut ett antal rader. Du kan också själv lägga in sannolikheter för de tretton matcherna.



HEMDATORER

Programmet är skrivet i GWbasic, så du behöver ingen översättning när du ska skriva in det på en PC-dator. Har du en VIC eller liknande så får du göra en del förändringar. Här kommer förändringarna.

Rad 20 tar du bort och rad 30 blir: 30 PRINT CHR$(147).

På samma sätt ändrar du raderna 110, 260 och 360.

Rad 160 ändrar du till: 160S=RND(0).

I rad 410 blir en liknande förändring, 410 P=INT (RND(0)★100)+.5.

I menyn ser du att du har tre alternativ:

"1. SLUMPADE MATCHER"
"2. EGNA SANNOLIKHETER"
"3. AVSLUTA"

Uppehåll

De slumpade matcherna har en rutin som ligger i raderna 110 till 250. Du får först svara på frågan om hur många matcher du vill ha. Därefter tar datorn ut antalet matcher och gör ett uppehåll vid var 15:e match. Rutinen för detta hittar du i rad 220. Glöm inte att skriva av matcherna.

De egna sannolikheterna startar i rad 260 och till rad 500. Rutinen för egna sannolikheter blir lite mer komplicerad än bara slumpade.

I raderna 290 till 310 lägger du in sannolikheterna i variablerna R(I,1), R(I,2) och R(I,3). Kom ihåg att skriva in talen som procenttal. I rad 320 kontrollerar datorn att summan blir 100%.

För att kunna jämföra med dina inslagna procenttal, får jag göra konstruktionen i rad 340, så att jag får ett värde för jämförelserna i raderna 420 till 440. Du ser att slutet på andra rutinen är lik rutinen för de slumpade tecknen.

Med egna sannolikheter kan du påverka varje match individuellt. Programmet tillåter dig också att sätta en sannolikhet till noll. Om du till exempel tror att en match blir en 1:a med 90% sannolikhet och ett X med 10% sannolikhet så kan du slå in detta med en nolla för sannolikheten för en tvåa.

I kommande artiklar kommer jag att skriva ett par program med filhantering. Det betyder att programmet lagrar information på diskett, hårddisk eller band.

Jag fick en fråga häromdagen om säkerhetskopiering till PC-datorer, huruvida man ska använda COPY eller BACKUP. Jag föredrar själv att använda COPY, men COPY har det problemet att den inte talar om när disketten är full. BACKUP kontrollerar om disketten blir full och säger till att du ska byta.

Jag har själv ett flertal gånger haft problem med BACKUP; därför har jag skaffat mig PCTOOLS som kontrollerar hur mycket disketten kan ta. Så numera använder jag COPY-funktionen i PCTOOLS och har bättre kontroll på mina säkerhetskopior.

Det har kommit två nya versioner av PCTOOLS. Akta er för den versionen som heter 5,0. Den har problem med COMPRESS-programmet. Jag förstörde alla filerna på min hårddisk. Köp istället version 5.5, som fungerar utan problem. Den nya versionen har ett menyskal, som är lätt att använda.

Jag avslutar med en förklaring på ett mycket vanligt fel på PC-program. Det händer då och då att man kommer till mig och säger: "Du, nu har jag kört det här programmet så många gånger och plötsligt så vill det inte längre. Vad har hänt?"

Min första fråga är: Har du skaffat ett nytt menyprogram? Om ja, så har vi troligtvis svarar på varför det gamla programmet inte fungerar.

Det finns i dag många program som lägger sig i minnet och då kommer i konflikt med andra program som också vill använda denna minnesarea. Det går i de flesta fall inte att ta bort dessa program; man får starta datorn igen för att programmet ska fungera.

```
10 DIM R(13,3)
20 RANDOMIZE(TIMER)
30 CLS
40 PRINT :PRINT TAB(10);"MENY"
50 PRINT :PRINT TAB(10);"1. SLUMPADE MATCHER"
60 PRINT :PRINT TAB(10);"2. EGNA SANNOLIKHETER"
70 PRINT :PRINT TAB(10);"3. AVSLUTA"
80 PRINT :PRINT TAB(10);"VAD öNSKAS";:INPUT Q
90 IF Q<1 OR Q>3 THEN 30
100 ON Q GOTO 110,260,510
110 CLS
120 PRINT :PRINT TAB(10);"HUR MÅNGA RADER";:INPUT N
130 PRINT :PRINT
140 FOR I=1 TO N
150 FOR J=1 TO 13
160 S=RND
170 IF S<.6 THEN PRINT "1 ";
180 IF S>=.6 AND S<.8 THEN PRINT "2 ";
190 IF S>=.8 THEN PRINT "X ";
200 NEXT J
210 PRINT
220 IF I/15=INT(I/15) THEN INPUT QQ$
230 NEXT I
240 PRINT :PRINT TAB(10);"TRYCK PÅ RETURN";:INPUT QQ$
250 GOTO 30
260 CLS
270 PRINT :PRINT TAB(10);"SLÅ IN DINA SANNOLIKHETER I PROCENT"
280 FOR I=1 TO 13
290 PRINT TAB(10);"MATCH NR";I;"SANNOLIKHET FöR 1";:INPUT R(I,1)
300 PRINT TAB(10);"MATCH NR";I;"SANNOLIKHET FöR 2";:INPUT  R(I,2)
310 PRINT TAB(10);"MATCH NR";I;"SANNOLIKHET FöR X";:INPUT  R(I,3)
320 T=R(I,1)+R(I,2)+R(I,3)
330 IF T<>100 THEN PRINT "MÅSTE BLI 100":GOTO 290
340 R(I,2)=R(I,1)+R(I,2)
350 NEXT I
360 CLS
370 PRINT :PRINT TAB(10);"HUR MÅNGA RADER";:INPUT N
380 PRINT :PRINT
390 FOR I=1 TO N
400 FOR J=1 TO 13
410 P=INT(RND*100)+.5
420 IF P<R(J,1) THEN PRINT "1 ";
430 IF P>=R(J,1) AND P<R(I,2) THEN PRINT "2 ";
440 IF P>=R(J,2) THEN PRINT "X ";
450 NEXT J
460 PRINT
470 IF I/15=INT(I/15) THEN INPUT QQ$
480 NEXT I
490 PRINT :PRINT TAB(10);"TRYCK PÅ RETURN";:INPUT QQ$
500 GOTO 30
510 END
```

Så kan du kolla personnummer

Register har jag skrivit om de senaste två veckorna. Naturligtvis blev då en fråga från Sven A i Malmö: "Register brukar man sätta i samband med personnummer. Kontrollerar datorn alltid personnumret och hur?"



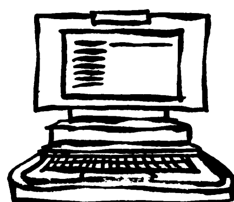
Christian Lindeberg

HEMDATORER

Svaret är ja, om du tänker på officiella register. I egna och privata register sker detta inte alltid. När det gäller register med personer får man inte glömma att man måste ha tillstånd från Datainspektionen för att för att föra ett register. Är man osäker skall man ta kontakt med Datainspektionen, Box 12050, 102 22 Stockholm eller per telefon 08/22 79 80.

Tio siffror

Sven frågade om personnummer. Ett personnummer består av tio siffror där de sex första är födelsetid. De nästa tre var i Sverige du är född. Den nionde om det är en pojke eller flicka. Flickor har jämna siffror och pojkar har udda. Tionde siffran är en kontrollsiffra på att det hela stämmer. Kontrollsiffran räknas ut med hjälpa av de andra nio. Uträkningen är ganska enkel och jag skall göra ett exempel och ett program. Jag använder mig av ett så kallat testnummer, som man kan få från Datainspektionen.



Personnumret är 590330—2783. Vi kan omedelbart se att personnumret tillhör en kvinna eftersom att nionde siffran är en jämn siffra. Den sista siffran är en trea och är kontrollsiffran. Nu skall vi kontrollera att detta stämmer. Uträkningen går till på följande sätt:

Man multiplicerar de nio siffrorna i personnumret varannan gång med två och varannan gång med ett. Därefter summerar man siffrornas värde, alltså inte talens värde. Nu tar man bort tiotalssiffran och drar denna ifrån talet tio och nu har vi fått fram kontrollsiffran.

Ett exempel

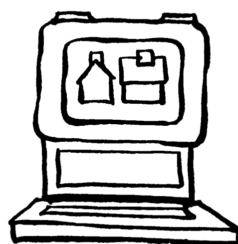
Det låter krångligt eller hur? Jag skall visa med ett exempel.

```
    5 9 0 3 3 0 2 7 8
  ★ 2 1 2 1 2 1 2 1 2
   10 9 0 3 6 0 4 7 16
```

Nu summerar vi siffrorna:
1+0+9+0+3+6+0+4+7+1+6=37. Nu tar vi bort tiotalssiffran som är 3 och nu får vi siffran 7 kvar. Denna siffra drar vi ifrån talet 10 och resultatet blir 3, vilket skall vara kontrollsiffran och detta ser vi att det stämmer med personnummer 590330—2783. Jag har gjort ett program där du också kan se uträkningarna. En sak måste jag påpeka att om sista siffran i uträkning blir en nolla så är också kontrollsiffran en nolla.

Bindestreck

En programteknisk del är naturligtvis hur man summerar siffrorna i talen 10 till 18. Det är ganska enkelt eftersom man tar entalsdelen och lägger till siffran 1. Tekniken man använder är att lägga siffrorna i strängar och sedan använda funktionerna VAL () och STR $(). Detta i kombination med de vanliga stränghanteringskommandona gör det ganska lätthanterligt. Kom ihåg att du slår in personnumren med ett bindestreck. Kontrollera gärna programmet med både riktiga och felaktiga personnummer. Lycka till!



```
10 CLS
20 PRINT "SLA IN PERSONNUMRET"
30 PRINT "ALLA SIFFROR OCH BINDESTRECK";
40 INPUT P$
50 PRINT :PRINT
60 A$=MID$(P$,1,6)+MID$(P$,8,3)
70 PRINT "PERSONNUMMER       ";
80 FOR I=1 TO 9
90 PRINT " ";MID$(A$,I,1);"  ";
100 NEXT I
110 PRINT:PRINT
120 PRINT "MULTIPLICERA MED ";
130 FOR I=1 TO 9
140 IF I/2=INT(I/2) THEN PRINT 1;" ";:GOTO 160
150 PRINT 2;" ";
160 NEXT I
170 PRINT:PRINT
180 PRINT "SVARET BLIR       ";
190 FOR I=1 TO 9
200 IF I/2=INT(I/2) THEN GOSUB 450:GOTO 220
210 GOSUB 470
220 PRINT N;" ";
230 T=T+N
240 NEXT I
250 PRINT:PRINT:PRINT "SUMMAN BLIR";
260 PRINT T
270 B$=STR$(T)
280 C$=MID$(B$,3,1)
290 R=VAL(C$)
300 K=10-R
310 IF K=10 THEN K=0
320 PRINT :PRINT "KONTROLLSIFFRAN BLIR 10 -";R;"=";K
330 PRINT :PRINT "KONTROLLSIFFRAN I PERSONNUMRET ÄR ";MID$(P$,11,1)
340 X$=STR$(K)
350 F=0
360 IF VAL(MID$(P$,11,1))=K THEN F=1
370 PRINT :PRINT
380 PRINT "PERSONNUMRET VAR ";P$
390 IF F=1 THEN PRINT:PRINT "OCH KONTROLLSIFFRAN ÄR RIKTIG":GOTO 440
400 IF F=0 THEN PRINT :PRINT "KONTROLLSIFFRAN ÄR FEL"
410 PRINT :PRINT "PERSONNUMRET BORDE VARA FÖLJANDE  ";
420 PRINT MID$(P$,1,10)+MID$(X$,2,1)
440 END
450 N=VAL(MID$(A$,I,1))
460 RETURN
470 IF 2*VAL(MID$(A$,I,1))>9 THEN N=2*VAL(MID$(A$,I,1))-9:GOTO 490
480 N=2*VAL(MID$(A$,I,1))
490 RETURN
```

Datorn kan bli en bra kassabok

Den här veckan skall jag visa ett program för en kassabok. Jag tror inte att man får bättre med pengar, men man får kontroll över var pengarna tar vägen även om detta kan vara deprimerande.



HEMDATORER

Innan jag kommer till programmet vill jag svara på en fråga om hårddiskar till PC-datorer. Frågan gällde hur man städar på en hårddisk. När man har använt sin hårddisk ett par månader, så har man både lagt till och tagit bort program.

Har man tagit bort många små program så blir det en massa tomma platser på hårddisken. När man sedan lägger dit ett större program så kommer detta inte att hamna i en följd utan lägger sig på de tomma platserna. Detta medför att hårddisken arbetar långsamt och att det sliter mera på disken än om programmet ligger i en följd. En bra regel är att man regelmässigt kör programmet CHKDSK.COM. När du kör det skall du skriva på följande sätt: CHKDSK/F

En varning

Detta kommando tar bort så kallade "lost clusters" på hårddisken. En bra investering är att köpa PCTOOLS ver 5.5 för att hålla rent och snyggt på sin hårddisk. I PCTOOLS 5.5 ingår ett program, som heter COMPRESS. Detta program städar upp och lägger dina filer i en sammanhängande kedja.

VARNING! Akta dig för PCTOOLS ver 5.0 för den innehåller ett par fel så att du kan förlora alla programmen på din hårddisk. Det hände mig med PCTOOLS 5.0.

Jag kör regelbundet med PCTOOLS 5.5 och den fungerar utan problem.

Nu över till kassaboken, som också lagrar alla transaktioner. I raderna 30 till 90 kan du se menyn, som hanterar inskrivning av intäkter och utgifter. Där finns också rutinerna för att spara och hämta uppgifter på intäkter och utgifter. För att kunna skilja på utgifter och inkomster har jag lagt till en stjärna på slutet av alla utgifter. Detta kan du se i rad 280. Denna använder jag sedan när jag läser in datan från min fil, som heter PENGAR.SEQ. Där känner programmet av om det finns en stjärna i slutet av ordet för då vet programmet att det är en utgift. Ändelsen SEQ lägger jag till för att jag skall se att det är en sekventiell fil och inte ett program. Ni som har VIC-64 och liknande får ändra i raderna 480, 560, 540 och 630 till följande.

480 OPEN 2,8,2, "PENGAR,S,W"
560 OPEN 2,8,2 "PENGAR,S,R"
540 CLOSE 2
630 CLOSE 2

En sak som du kan fundera på är hur får man bort stjärnan så att den inte syns i översikten med transaktioner, men den skall fortfarande finnas kvar när jag lagrar mina utgifter. Tänk på det till nästa vecka.

```
10 DIM IN(1000),KO(1000),IN$(1000),KO$(1000)
20 CLS
30 PRINT :PRINT TAB(10);"KASSABOK"
40 PRINT :PRINT TAB(10);"1. INTÄKTER"
50 PRINT :PRINT TAB(10);"2. UTGIFTER"
60 PRINT :PRINT TAB(10);"3. VISA TRANSAKTIONER"
70 PRINT :PRINT TAB(10);"4. HÄMTA REGISTER"
80 PRINT :PRINT TAB(10);"5. AVSLUTA OCH SPARA REGISTER"
90 PRINT :PRINT TAB(10);"VAD ÖNSKAS";
100 INPUT Q
110 IF Q<1 OR Q>5 THEN 20
120 ON Q GOTO 130,240,350,560,480
130 CLS
140 N=N+1
150 PRINT :PRINT TAB(10);"INTÄKTER"
160 PRINT :PRINT TAB(10);"SKRIV TYP AV INTÄKT";
170 INPUT IN$(N)
180 PRINT :PRINT TAB(10);"HUR STOR ÄR SUMMAN";
190 INPUT IN(N)
200 PRINT :PRINT TAB(10);"SKALL DU SLÅ IN FLER";
210 INPUT QQ$
220 IF QQ$="j" OR QQ$="J" THEN 130
230 GOTO 20
240 CLS
250 N=N+1
260 PRINT :PRINT TAB(10);"UTGIFTER"
270 PRINT :PRINT TAB(10);"SKRIV TYP AV UTGIFT";
280 INPUT KO$(N):KO$(N)=KO$(N)+"*"
290 PRINT :PRINT TAB(10);"HUR STOR ÄR SUMMAN";
300 INPUT KO(N)
310 PRINT :PRINT TAB(10);"SKALL DU SLÅ IN FLER";
320 INPUT QQ$
330 IF QQ$="j" OR QQ$="J" THEN 240
340 GOTO 20
350 CLS
360 PRINT :PRINT TAB(10);"VISA TRANSAKTIONER OCH SALDO"
370 PRINT :PRINT
380 PRINT "TRANSAKTION";TAB(20);"SUMMA";TAB(30);"SALDO"
390 FOR I=1 TO N
400 IF IN(I)>0 THEN PRINT IN$(I);TAB(20);IN(I);TAB(30);
410 IF KO(I)>0 THEN PRINT KO$(I);TAB(20);KO(I);TAB(30);
420 SUM=SUM+IN(I):MIN=MIN+KO(I)
430 IF KO(I)<>0 OR IN(I)<>0 THEN PRINT SUM-MIN
440 IF I/15=INT(I/15) THEN PRINT :PRINT TAB(10);:INPUT QQ$
450 NEXT I
460 PRINT :PRINT TAB(10);"TRYCK PÅ RETURN";:INPUT QQ$
470 GOTO 20
480 OPEN"O",#1,"PENGAR.SEQ"
490 PRINT#1,N
500 FOR I=1 TO N
510 IF IN(I)>0 THEN PRINT#1,IN$(I):PRINT#1,IN(I)
520 IF KO(I)>0 THEN PRINT#1,KO$(I):PRINT#1,KO(I)
530 NEXT I
540 CLOSE 1
550 END
560 OPEN"I",#1,"PENGAR.SEQ"
570 INPUT#1,N
580 FOR I=1 TO N
590 INPUT#1,T$,T
600 IF RIGHT$(T$,1)="*" THEN KO$(I)=T$:KO(I)=T:GOTO 620
610 IN$(I)=T$:IN(I)=T
620 NEXT I
630 CLOSE 1
640 GOTO 20
```

Här är ett svårare matematikprogram

I dag fortsätter vi med matematikprogrammen. Hoppas att du har tränat på förra veckans program. Då skulle ni nå fram till ett tal utan att få addera eller minska er dit.



HEMDATORER

Den här gången är det ett program, som blir lite svårare. Du får slutmålet och det värde du skall starta med. Nu får du använda vilket räknesätt du vill men bara med talet 2. Datorn räknar antalet gånger du försöker. Du bör klara vilket tal som helst under 10 gånger. Jag skall ta ett exempel:

Du får ett slutmål på 96 och din start är 32. Nu kan du multiplicera med 2, addera med 2, minska med 2 eller dividera med två. Slutresultatet skall vara 96.

Jag börjar och dividerar med 2, resultatet blir 15.

Nu minskar jag med 2 och får 14. Efter ytterligare en minskning så hamnar jag på 12.

Nu multiplicerar jag en gång och får 24. En ny multiplikation ger 48 och en sista multiplikation ger svaret 96. Jag klarade det på 6 gånger, vilket är godkänt.

Mellan 1 och 100

Programmet ger ett tal mellan 1 och 100. Du kan öka svårighetsgraden genom att öka till 1.000 eller högre.

Här kommer programmet:

```
10 CLS
20 RANDOMIZE (TIMER)
30 N=INT(RND*100)
40 PRINT:PRINT TAB(10);"TANKEVERKSAMHET"
50 PRINT:PRINT TAB(10);"DITT SLUTMÅL ÄR TALET";N
60 V=INT(RND*N/2)
70 PRINT:PRINT TAB(10);"DU SKALL BÖRJA MED VÄRDET";V
80 PRINT:PRINT TAB(10);"VILKET RÄKNESÄTT VILL DU HA"
90 T=T+1
100 PRINT:PRINT TAB(10);"1. ADDITION MED 2"
110 PRINT:PRINT TAB(10);"2. SUBTRAKTION MED 2"
120 PRINT:PRINT TAB(10);"3. MULTIPLIKATION MED 2"
130 PRINT:PRINT TAB(10);"4. DIVISION MED 2"
140 PRINT:PRINT TAB(10);"VAD ÖNSKAR DU";
150 INPUT Q
160 IF Q<1 OR Q>5 THEN 80
170 ON Q GOSUB 190,240,290,340
180 GOTO 90
190 V=V+2
200 IF V=N THEN 390
210 CLS:PRINT:PRINT TAB(10);"DITT SLUTMÅL ÄR TALET";N
220 PRINT:PRINT TAB(10);"DITT VÄRDE ÄR NU";V
230 RETURN
240 V=V-2
250 IF V=N THEN 390
260 CLS:PRINT:PRINT TAB(10);"DITT SLUTMÅL ÄR TALET";N
270 PRINT:PRINT TAB(10);"DITT VÄRDE ÄR NU";V
280 RETURN
290 V=V*2
300 IF V=N THEN 390
310 CLS:PRINT:PRINT TAB(10);"DITT SLUTMÅL ÄR TALET";N
320 PRINT:PRINT TAB(10);"DITT VÄRDE ÄR NU";V
330 RETURN
340 V=INT(V/2+.5)
350 IF V=N THEN 390
360 CLS:PRINT:PRINT TAB(10);"DITT SLUTMÅL ÄR TALET";N
370 PRINT:PRINT TAB(10);"DITT VÄRDE ÄR NU";V
380 RETURN
390 PRINT:PRINT
400 PRINT:PRINT TAB(10);"GRATULERA DU KLARADE DET PÅ"
410 PRINT:PRINT TAB(10);T;"GÅNGER"
```

Programmet bygger på fyra stycken subrutiner. Du väljer i rad 140 och 150. I rad 170 hoppar datorn till den subrutin du valt. Om du har valt en etta så hamnar du i rad 190 och har du valt en tvåa så hamnar du i rad 240.

Subrutinen utför nu den matematiska beräkning, som du har bestämt med ditt val. Efter beräkningen testar programmet om du har uppnått målet eller om du skall fortsätta med fler beräkningar.

Den basic jag använder är GW-Basic och använder du en annan basic så titta i instruktionsboken vad du skall använda i stället för mina uttryck.

Jag skall visa på en del funktioner, som kan skilja.

CLS = rensa skärmen

RANDOMIZE(TIMER) = startar en speciell slumpvalsgenerator.

RND = slumpvalsgeneratorn som ger ett tal till din variabel.

Dessa är de funktioner, som kan skilja med andra basicdialekter.

Kopieringar

Nu till en brevfråga: Thure L i Lund, som har en fråga om MS-DOS, alltså operativsystemet till PC-datorer. Frågan gäller kommandot XCOPY och vad som skiljer detta från COPY.

Om du använder COPY så kopierar du ett program från en disk till en annan diskenhet.

Med XCOPY har du ett kommando, som liknar BACKUP-kommandot om man tittar på de parametrar, som man kan använda.

Den mest använda parametern är /s. Den gör att du kopierar även det som finns i underbiblioteken. Jag skall ta ett exempel.

Du har ett bibliotek med namnet KALKYL, som har två underbibliotek, SKATT och PENGAR. Nu skall du kopiera allt under biblioteket KALKYL med underbiblioteken SKATT och PENGAR. Med XCOPY kan du skriva på följande sätt om du vill kopiera från C: till A:.

XCOPY C: KALKYL A: /s

Nu kopieras även underbiblioteken in på A:

Lycka till!

Gör egna julkort med din hemdator

Nu börjar vi närma oss julen. Det är i dag exakt en månad till julafton. Så därför har jag gjort ett litet program, som kan rita ut julkort och skriva ut etiketter.



HEMDATORER

Ett fel blev det i förra veckans program i rad 10 hade jag tappat bort en dimensionering. Rad 10 skall se ut så här:
10 DIM N(27,7), R(27), T(27)

Om man inte har med den sista delen med T(27), så blir det fel i rad 290.

Nu skall programmet fungera.

Tack Conny O. Lund för att du snabbt uppmärksammade mig på problemet.

Ett varmt tack till signaturen Y.L. för ditt trevliga brev.

I den här spalten finns inga onödiga frågor utan bara frågor. Jag vet att många sitter hemma och har en hel del frågor, men tycker att kanske frågorna är för enkla. Svaret är alltid enkelt när man kan. Men en fråga, som jag inte vet svaret på är alltid en svår fråga. Programmering är också ett svårare avsnitt än något annat här i livet. Eftersom den allra minsta detalj kan göra att ett program inte fungerar. Skriver jag ett brev så spelar det inte så stor roll om jag skriver lite fel eller glömmer ett kommatecken. Men i programmering kan ett glömt litet tecken få förödande konsekvenser.

Byta bibliotek

Jag fick ett brev från Åke A i Landskrona, som undrar om man på en PC kan byta bibliotek i ett basicprogram? Svaret är ja. Man kan byta bibliotek, göra nya underbibliotek och ta bort bibliotek. Jag rekommenderar bara att använda funktionen för att byta bibliotek, därför att man har en dålig överblick i basic. Skall man skapa nya eller ta bort bibliotek så gör detta från DOS-miljön och inte från ett basicprogram.

Jag har skrivit ett litet program som demonstrerar hur det fungerar. Jag har även med funktionen BEEP, som kan väcka sömniga personer. Programmet fungerar bara i basic på en PC.

För GW-basic

Här kommer programmet:

```
10 CLS
20 BEEP
30 PRINT "ATT SKIFTA BIBLIOTEK"
40 CHDIR "C:/ARBETET"
50 FILES
60 BEEP
70 STOP
80 CHDIR "C:/BASIC"
90 FILES
100 BEEP
```

Julkortsprogrammet innehåller inga konstigheter. Programmet är skrivet för GW-basic, så om du har en VIC-64 eller liknande bör du ändra alla LPRINT till PRINT och istället ha en OPEN 4,4 innan utskriften och ett CLOSE 4 efter utskriften.

Lycka till!

```
10 CLS
20 PRINT :PRINT TAB(10);"MENY"
30 PRINT :PRINT TAB(10);"FÖR JULHÄLSNINGAR"
40 PRINT :PRINT TAB(10);"1. SKRIVA UT ADRESSLAPPAR"
50 PRINT :PRINT TAB(10);"2. SKRIVA UT JULKORT"
60 PRINT :PRINT TAB(10);"3. AVSLUTA"
70 PRINT :PRINT TAB(10);"VAD ÖNSKAS";
80 INPUT Q
90 ON Q GOTO 110, 300,680
100 IF Q>3 OR <1 THEN 10
110 CLS:RESTORE
120 PRINT :PRINT TAB(10);"UTSKRIFT AV ADRESSLAPPAR"
130 PRINT :PRINT TAB(10);"GLÖM INTE ATT STARTA SKRIVAREN"
140 PRINT :PRINT TAB(10);"NÄR DU ÄR KLAR SKALL DU"
150 PRINT :PRINT TAB(10);"TRYCKA PÅ RETURN";
160 INPUT QQ$
170 REM
180 READ N$(1),F$(1),AD$(1),PN$(1),PO$(1)
190 IF N$(1)="-1" THEN 280
200 READ N$(2),F$(2),AD$(2),PN$(2),PO$(2)
210 IF N$(2)="-1" THEN 280
220 LPRINT :LPRINT
230 LPRINT TAB(10);N$(1);" ";F$(1);TAB(50);N$(2);" ";F$(2)
240 LPRINT TAB(10);AD$(1);TAB(50);AD$(2)
250 LPRINT TAB(10);PN$(1);"  ";PO$(1);TAB(50);PN$(2);"  ";PO$(2)
260 LPRINT :LPRINT
270 GOTO 170
280 REM
290 GOTO 10
300 CLS:RESTORE
310 PRINT :PRINT TAB(10);"UTSKRIFT AV JULKORT"
320 PRINT :PRINT TAB(10);"HUR MÅNGA VILL DU HA";:INPUT N
330 PRINT :PRINT TAB(10);"GLÖM INTE ATT STARTA SKRIVAREN"
340 PRINT :PRINT TAB(10);"NÄR DU ÄR KLAR SKALL DU"
350 PRINT :PRINT TAB(10);"TRYCKA PÅ RETURN";
360 INPUT QQ$
370 REM
380 FOR I= 1 TO N
390 LPRINT :LPRINT
400 LPRINT "*****************************************************"
410 LPRINT "*                                                   *"
420 LPRINT "*                                                   *"
430 LPRINT "*          *      *    * *       *  *  *  *         *"
440 LPRINT "*        *      *   *  *   *     *  *  *  *         *"
450 LPRINT "*        *  **  *   *  *   *   * *  *  *  *         *"
460 LPRINT "*         **      *    * *     * *  ****  ****      *"
470 LPRINT "*                                                   *"
480 LPRINT "*                                                   *"
490 LPRINT "*                 Ö N S K A R                       *"
500 LPRINT "*                                                   *"
510 LPRINT "*                                                   *"
520 LPRINT "*                                                   *"
530 LPRINT "*                                                   *"
540 LPRINT "*  ***********************************************  *"
550 LPRINT "*                                                   *"
560 LPRINT "*
570 LPRINT "*  ***********************************************  *"
580 LPRINT "*                                                   *"
590 LPRINT "*                                                   *"
600 LPRINT "*  ***********************************************  *"
610 LPRINT "*                                                   *"
620 LPRINT "*                                                   *"
630 LPRINT "*****************************************************"
640 LPRINT :LPRINT :LPRINT
650 NEXT I
660 REM
670 GOTO 10
680 END
690 DATA ARBETET,,BOX 125,201 21,MALMÖ
700 DATA LINDEBERG,CHRISTIAN,BOX 6,236 00,HÖLLVIKEN
710 DATA -1,-1,-1,-1,-1
```

Julrim på datorn

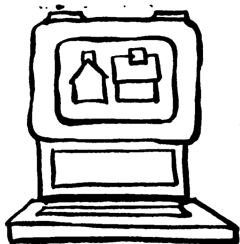
När julen närmar sig brukar en hel del börja att tänka på julrim till julklapparna. Vi som har datorer kan vara lite mera avancerade och låta datorn skriva en liten dikt till paketen. Ni som har varit med länge kommer säkert ihåg HAIKU-dikterna, som jag har haft i tidigare artiklar.

Vad är en HAIKU-dikt? Enligt Stora Focus: "haiku, japansk treradig versform med sammanlagt 17 stavelser, fördelade enligt schemat 5-7-5.

Haikupoesin hade sitt ursprung från 1600-talet då familjen Tokugawa 1603 erövrade shoguntiteln och den egentliga makten. Städerna och borgarna fick större betydelse i kulturlivet, med tryckkonstens utbredning, mer folkliga teaterformer och haikupoesins blomstring."

Exakthet

Haikudikten lämpar sig väl för datorer genom sin exakthet.



Här kommer en Haikudikt:

Tomten skrattande
under en stilig julgran
en fridfull juldag

Om du tittar på dikten ovan så hittar du DATA-satser i sju olika rader.

Om vi ser närmare på dessa rader i dikten intill så hittar du tekniken med dikterna.

I dikten ser du att jag tagit ett ord från varje DATA-rad.

I rad 220 och rad 310 har jag ordet "EN". Tänk på att det måste finnas lika många ord på varje rad.

Antalet ord styr du med slumptalsraderna 30 till 90. Om du använt siffran sex, så måste du ha sex ord i varje DATA-rad. Du kan också använda ett ord flera gånger, som jag har gjort i rad 200. I varje loop med READ, så tar datorn ut ett ord beroende på den siffra datorn har fått i raderna 30 till 90.



När programmet har skrivit ut dikten i raderna 400 till 420, så gör programmet en RESTORE i rad 430. Detta för att datorn skall kunna använda ordet igen till nästa dikt.

Sista delen av programmet skrivet ut dikterna i den ordning orden står i datasatserna.

Skriva ut

Om du har en skrivare, så kan du låta programmet skriva ut dem direkt genom att ändra i raderna 400 till 420 till LPRINT om du skriver en GWbasic liknande dialekt. Har du en VIC eller liknande kan du använda kommandot OPEN 4,4 på en rad 395 och avsluta med ett CLOSE 4 på en rad 425.

Lycka till med dikterna!

Ett fel

I veckan fick jag ett brev från Conny O i Lund. Conny har problem med ett kalorieberäkningsprogram, som mycket liknar en del av förra veckans bantningsprogram. Jag kan inte hitta mer än ett fel i ditt program och det är rad 220. Du får inte ha ett mellanslag i variabeln KAL (L). Detta är anledningen till att det blir "Totalt 0 kalorier". Det andra felet tror jag beror på när du slår in namnet på dina födoämnen. Jag skall visa med ett exempel.

Du har en DATA-rad, som ser ut så här:

410 DATA Glass,225,Mjölk,62,Smör,748

Faran att använda skrivsättet med både stora och små bokstäver är att du måste komma ihåg hur du skrev i DATA-raden. Om du nu söker på SMÖR eller smör, så kommer datorn inte att hitta födoämnet eftersom du i DATA-satsen har stavat ordet Smör. Om Du tittar på mina program så använder jag alltid stora bokstäver (VERSALER), för att jag inte skall göra fel. Annars måste du lägga in alla kombinationer i DATA-satserna. Då skulle rad 410 se ut så här:

410 DATA Glass,225,GLASS, 225,glass,225 o s v

Detta är ett lite onödigt arbete.

En annan sak, som är mycket viktig, i DATA-satser är att man håller kontroll på sina kommatecken. Om du råkar få ett kommatecken i slutet på en rad så läser datorn detta som om det fanns tecken efter kommatecknet. Datorn kommer då att lagra en tom sträng eller siffran 0 om det är en numerisk variabel man läser med READ-satsen.

Lycka till med programmeringen.



HEMDATORER

```
120 DATA TOMTEN,MÅNEN,SOLEN,BARNEN,JULEN,MAMMA
160 DATA SUCKANDE,BUGANDE,SKRATTANDE,LEENDE,KOMMANDE,KRAMMANDE
200 DATA OVAN,BAKOM,UNDER,ÖVER,ÖVER,GENOM
250 DATA KNOTIG,SLITEN,GYLLNE,STILIG,BLÅSIG,GLAD
290 DATA KÄRVE,SPILTA,LAMPA,LEKPLATS,HAVSSTRAND,JULGRAN
340 DATA SÅDAN,KULEN,FRIDFULL,TIDIG,SANDIG,STJÄRNKLAR
380 DATA HÖSTKVÄLL,VINTERNATT,KVÄLLSTUND,LUCIAMORGON,JULDAG,ADVENT
```

● *Observera att du måste skriva dikten utan mellanslag och rad för rad.*




```
10 CLS
20 RANDOMIZE(TIMER)
30 B=INT(RND*6+1)
40 C=INT(RND*6+1)
50 D=INT(RND*6+1)
60 F=INT(RND*6+1)
70 G=INT(RND*6+1)
80 J=INT(RND*6+1)
90 K=INT(RND*6+1)
100 FOR I=1 TO 6
110 READ B$(I)
120 DATA TOMTEN,MÅNEN,SOLEN,BARNEN,JULEN,MAMMA
130 NEXT I
140 FOR I=1 TO 6
150 READ C$(I)
160 DATA SUCKANDE,BUGANDE,SKRATTANDE,LEENDE,KOMMANDE,KRAMMANDE
170 NEXT I
180 FOR I=1 TO 6
190 READ D$(I)
200 DATA OVAN,BAKOM,UNDER,ÖVER,ÖVER,GENOM
210 NEXT I
220 E$="EN"
230 FOR I=1 TO 6
240 READ F$(I)
250 DATA KNOTIG,SLITEN,GYLLNE,STILIG,BLÅSIG,GLAD
260 NEXT I
270 FOR I=1 TO 6
280 READ G$(I)
290 DATA KÄRVE,SPILTA,LAMPA,LEKPLATS,HAVSSTRAND,JULGRAN
300 NEXT I
310 H$="EN"
320 FOR I=1 TO 6
330 READ J$(I)
340 DATA SÅDAN,KULEN,FRIDFULL,TIDIG,SANDIG,STJÄRNKLAR
350 NEXT I
360 FOR I=1 TO 6
370 READ K$(I)
380 DATA HÖSTKVÄLL,VINTERNATT,KVÄLLSTUND,LUCIAMORGON,JULDAG,ADVENT
390 NEXT I
400 PRINT:PRINT TAB(10);B$(B);" ";C$(C)
410 PRINT TAB(10);D$(D);" ";E$;" ";F$(F);" ";G$(G)
420 PRINT TAB(10);H$;" ";J$(J);" ";K$(K)
430 RESTORE
440 PRINT:PRINT TAB(10);"VILL DU HA EN NY DIKT (J/N)";
450 INPUT A$
460 IF A$="J" THEN 10
470 PRINT:PRINT TAB(10);"TACK FÖR ATT DU VILLE LÄSA MINA DIKTER"
480 CLS:PRINT:PRINT TAB(10);"ORIGINALDIKTERNA"
490 FOR I=1 TO 6
500 PRINT B$(I);" ";C$(I)
510 PRINT D$(I);" ";E$;" ";F$(I);" ";G$(I)
520 PRINT H$;" ";J$(I);" ";K$(I)
530 PRINT:PRINT:INPUT "TRYCK PÅ RETURN";Q$
540 PRINT:PRINT
550 NEXT I
560 END
```

Nya program

Nu är det tid att slå sig ner vid datorerna igen. Jag har använt sommaren att mellan baden testa en del nya program.



HEMDATORER

Jag började med MS-DOS 5.0 från Microsoft och sedan med IBM's DOS 5.0. Skillanden mellan de två operativsystemen var att MS-dos 5.0 var en så kallad "upgrade", så att man startade inte datorn från diskett 1, utan man installerade den nya dosen genom att starta som vanligt och sedan från ett SETUP-program från diskett 1 så installerades den nya dosversionen. IBM's DOS var en vanlig installation så att man startade upp datorn på diskett 1.

Bland nyheterna fanns en riktig editor, så att alla som inte trivs med EDLIN kan nu använda programmet EDIT i stället. Programmet Edit är egentligen editorn till Qbasic, som man också får med till programmet.

Qbasic är den nya typen av basic där du kan programmera utan att använda radnummer. Man delar in programmet i rutiner, som man sedan anropar. Den vanliga GWbasic finns också med, till IBM heter den IBM basic.

Installation

En sak är viktig att tänka på. Du kan inte installera IBM's DOS 5.0 på annat än IBM datorer. Men Microsoft's MS-DOS 5.0 kan du installera på alla datorer. IBM's DOS förutsätter att du har IBM's speciella ROM-kapsel i datorn för bland annat BASIC och QBasic skall fungera. Fungerar inte QBasic så fungerar inte editorn EDIT, vilken egentligen är en del av QBasic.

Minneshanteringen har blivit klart förbättrad, man kan få ut cirka 630 Kb av de 640 Kb man har från början. Skall man kunna utnyttja DOS 5.0 bör man ha minst en 80286 processor och ett internminne på minst 2 Mb och hårddisk.

På sekundärminnessidan kan DOS 5.0 hantera hårddiskar på upp till 2 Gb (1 Gigabytes = 1000 MB = 1 000 000 000 bytes) och de nya diskettstationerna, som klarar av 3,5 tums disketter på 2,88 Mb lagringskapacitet.

DOS 5.0 har också en inbyggd hjälp så att man kan skriva kommandot följt av/? så får man en hjälp på skärmen.

DOS 5.0 innehåller mycket, men uppfyller du inte kraven ovan så behöver du byta till DOS 5.0.

Egna program

Under sommaren så har jag också testat ett nytt Basic-program. Programmet heter Visual Basic och kommer från Microsoft. Programmet går bara att köras under Windows. Programmet är mycket lättarbetat och en stor fördel är att det innehåller en kompilator så att man gör egna EXE-filer av sina program. Eftersom programmet arbetar med och under Windows så kan man göra verkligt professionella layouter.

En förklaring till ordet kompilator. Man skiljer på två olika typer av programmeringsspråk. Det ena interpreterande och det andra är kompilerande språk. När vi kör våra basicprogram måste vi först ladda in vår basic och sedan när vi skriver RUN, så översätter basictolken vårt program till maskinspråk och vårt program kör. Basic är ett typiskt interpreterande språk.

Om vi jämför med Pascal så är det ett kompilerande språk. För att Pascal gör ett maskinspråksprogram av Pascalprogrammet så att man kan köra sitt Pascalprogram direkt i datorn och behöver inte ladda in sitt Pascalprogram först.

Till en hel del av de moderna Basic-dialekterna finns i dag kompilatorer. Alltså ett program som överför basicspråket till maskinspråk, vilket gör att man inte behöver ladda in basicprogrammet innan man skall köra sitt program. En fördel till med en kompilator är att ingen kan gå in och göra något i programmet eftersom det ligger i maskinspråk. Visual basic har en sådan kompilator i sitt program.

Bra investering

Har du Windows 3.0 till din dator så är det en bra investering att köpa Visual Basic. Visual Basic använder inga radnummer utan har etiketter på rutinerna. Fördelen med denna typ av programmering är att den blir lättare att strukturera. Det är det vanligaste problemet med att programmera i Basic att programmen blir ganska röriga. Till er som vill börja med QBASIC eller VISUAL BASIC rekommenderar jag att låna boken "Programmera med Struktur" av Sven Burman och Bo Bäckman. Förlaget är Liber och ISBN-nummer är 91-40-80072-5. Tillbaks till Visual Basic. Visual Basic har också ett ikon-bibliotek, som innehåller 400 st ikoner (ikon är en liten symbolbild). Dessa ikoner kan du använda i dina basicprogram, alla är i 16 färger. Du kan också göra egna ikoner i den medföljande ikoneditorn.

Du kan ta in bilder som du har gjort i Windows Paintbrush, men bilderna måste då ha extensionen. BMP.

En ovanlig detalj är att Visual Basic kan arbeta med DDE (Dynamic Data Exchange). DDE är en mekanism, som understödjs i Windows. Om jag har till exempel kalkylprogrammet Excel kan jag med ett program skrivet i Visual Basic koppla detta till en kalkyl som jag har gjort i Excel. Det finns många Windows-applikationer, som kan använda DDE.

I Visual Basic kan man göra egna rullgardinsmenyer, vilket ger ett mycket elegant intryck. De applikationer du gör i Visual Basic och kompilerar är du fri att sälja.

Det finns mycket mera att säga om Visual Basic, men jag återkommer med mera i kommande artiklar.

Jag har fått en del brev i sommar och bland dem var ett från Pia K i Malmö. Pia ville ha ett litet program, som hon kunde använda som kontroll vid bantning.

Jag har gjort ett litet program, som också kontrollera om du är för tjock. Förut räknade man med längden minus 100 och det som blev över skulle sen överensstämma med vikten. Nu använder man i stället en annan formel där man tar vikten i kilo och dividerar med kvadraten på längden i meter. Värdet man får fram skall vara under 25 för att inte vara överviktig. Är värdet mellan 25 och 30 är man överviktig och är det över 30 är man i riskzonen för olika välfärdssjukdomar.

Jag har skrivit programmet med förklarande REM-satser. Lycka till Pia!

Ha en bra vecka!

Bantningsprogram

```
10 REM **********************INMATNING AV VIKT
20 CLS
30 PRINT : PRINT
40 PRINT "BANTNINGSPROGRAM "
50 PRINT
60 PRINT "NU KOMMER EN DEL FRÅGOR"
70 PRINT "SOM DU SKALL SVAR PÅ"
80 PRINT
90 PRINT "HUR MYCKET VÄGER DU";
100 INPUT KG
110 PRINT "HUR LÅNG ÄR DU ANGE I METER";
120 INPUT L
130 REM ********************UTRÄKNING AV TJOCKVÄRDE
140 TJ = KG / (L) ^ 2
150 IF TJ < 25 THEN D$ = "DU HAR INGET ATT OROA DIG FÖR"
160 IF TJ > 25 AND TJ < 30 THEN D$ = "DU BÖR NOG VÄGA LITE MINDRE"
170 IF TJ > 30 THEN D$ = "SÄTT OMEDELBART IGÅNG MED ATT BANTA"
180 REM ********************AKTIVITET
190 PRINT
200 PRINT "HUR AKTIV ÄR DU?"
210 PRINT
220 PRINT "1. STILLASITTANDE"
230 PRINT "2. NÅGOT AKTIV"
240 PRINT "3. MÅTTLIGT AKTIV"
250 PRINT "4. MYCKET AKTIV"
260 PRINT
270 INPUT "SKRIV SIFFRAN FÖR AKTIVITETSGRAD"; AK
280 IF AK = 1 THEN AK = 31
290 IF AK = 2 THEN AK = 33
300 IF AK = 3 THEN AK = 35
310 IF AK = 4 THEN AK = 37
320 REM****************** UTRÄKNING AV KALORIEBEHOV
330 KAL = AK * KG
340 REM ***************** KALORIEBEHOV PER DAG
350 PRINT
360 PRINT "DITT KALORIBEHOV PER DAG"
370 PRINT "FÖR ATT BEHÅLLA DIN VIKT ÄR ="; KAL
380 REM***************** BANTNING
390 PRINT
400 PRINT "HUR MYCKET VILL DU GÅ NER PER VECKA"
410 PRINT "UTTRYCKT I KG",
420 INPUT NE
430 MIN = NE * 1000
440 REM****************** UTSKRIFT AV BANTNINGFÖRESKRIFTER
450 PRINT
460 PRINT "FRÅN OCH MED NU FÅR DU BARA"
470 PRINT "ÄTA FÖLJANDE ANTAL KALORIER"
480 PRINT "PER VECKA FÖR ATT GÅ NER"
490 PRINT NE; "KG PER VECKA"
500 PRINT : PRINT
510 PRINT "VECKA 1."; KAL - MIN; "KALORIER"
520 KG = KG - 1: KAL = AK * KG
530 PRINT "VECKA 2."; KAL - MIN; "KALORIER"
540 KG = KG - 1: KAL = AK * KG
550 PRINT "VECKA 3."; KAL - MIN; "KALORIER"
560 KG = KG - 1: KAL = AK * KG
570 PRINT "VECKA 4."; KAL - MIN; "KALORIER"
580 KG = KG - 1: KAL = AK * KG
590 PRINT "VECKA 5."; KAL - MIN; "KALORIER"
600 PRINT : PRINT " DITT VÄRDE BLEV ="; TJ; " "; D$
```

Bantningsprogram



HEMDATORER

Det är trevligt att vara tillbaka. Förra veckans nummer hann knappt komma i brevlådan innan telefonen ringde. Sommaren har tydligen varit matrik eftersom man behövde ett bantningsprogram.

En av dem som ringde, som ville vara anonym, hade för sig att jag något tidigare år hade haft ett bantningsprogram med en inbyggd kaloritabell.

Det är alldeles riktigt och jag har skrivit om förra veckans program och lagt till en kaloritabell. Tabellen kan du sedan utvidga själv. Men tänk på att de två sista raderna måste du flytta neråt hela tiden för att de måste alltid ligga sist.

Du börjar alltså nästa datarad på rad 1190 och skriver in födoämnets namn och sedan efter kommatecknet hur många kalorier det innehåller. Ha inte mer än tre födoämnen på varje rad. När du har kompletterat listan så avslutar du med två rader -1,-1 precis som raderna 1190 och 2000.

Programmet kan förskönas på många punkter men det tar i så fall så stor plats. Till nästa gång skall jag byta ut kaloridelen och lägga till en sorteringsrutin.

Det har kommit en del frågor om program till PC-datorer, som kan vara av praktisk nytta. Jag har titta på två intressanta program. Det ena heter PCGlobe och är en "elektronisk atlas". Programmet innehåller kartor, grafik, fakta och tabeller för mer än 190 länder. Det enade Tyskland finns också med men man kan också få uppgifter om Tyskland som öst och väst. De största städerna har även klimattabeller, så att man kan se temperaturförändringen över året. Till varje land finns flagga och nationalsång.

Programmet innehåller också en stor databas med uppgifter om hälsovård, skola, ekonomi och mycket, mycket annat. Det finns i programmet också en automatisk valutaomvandlare, som omvandlar valutor från olika länder till svensk eller någon annan valuta. Internationella telefonkoder och telexkoder och beräkningar mellan olika platser genom att bara peka på dem.

Det finns också uppgifter om styrelseform, ledare och politiska partier för alla länderna.

En annan intressant sak med programmet är att man kan exportera både grafik och text till andra program. Detta är en bra sak när man skall skriva egna saker för då kan man hämta fakta direkt från programmet.

Jag skulle sammanfatta programmet, som mycket bra både för dig som går i skolan och för alla som vill vara allmänbildade. Ett bra sätt att snabbt skaffa sig information om olika länder.

Det andra programmet, som jag har tittat på är EZCosmos. Detta är ett astronomiprogram. Det mest fantastiska, som jag har sett i denna klass. Du kan studera himlen med alla planeter och stjärnor. Du kan själv bestämma om du vill stå i Malmö eller i Sydney eller vilken plats som helst på jorden. Hela stjärnhimlen kan animeras så att jag kan bestämma hur stjärnhimlen såg ut 1986 i januari till mars samma år i Peking. Jag bestämmer själv tidsintervallet för animeringen, så det kan vara ett år mellan bilderna eller en timme. Man kan även välja om månen skall animeras på samma gång. Månen visas då i sin rätta fas. Jag är ingen amatörastronom, men jag blev mycket fascinerad av hela programmet. För dig som är intresserad av det som finns utanför jordklotet är detta ett mycket bra program.

Lycka till med bantningsprogrammet och ha en bra vecka!

```
10 CLS
20 DIM F$(25),V(25),KAL(25),TOT(25)
30 CLS:PRINT:PRINT:PRINT "     M E N Y"
40 PRINT :PRINT "1. HUR MYCKET SKALL DU VÄGA"
50 PRINT :PRINT "2. HUR MÅNGA KALORIER INNEHÅLLER MIN MAT"
60 PRINT :PRINT "3. AVSLUTA"
70 PRINT :PRINT "VAD ÖNSKAS";:INPUT Q$
80 ON VAL(Q$) GOTO 450,100,930
90 GOTO 30
100 CLS:PRINT :PRINT
110 PRINT "KALORIEBERÄKNINGSPROGRAM"
120 PRINT:PRINT"VILL DU HA EN LISTA ÖVER ALLA ÄMNEN (J/N)";:INPUT Q$
130 IF Q$="J" OR Q$="j" THEN 360
140 PRINT:PRINT "HUR MÅNGA FÖDOÄMNEN"
150 PRINT "VILL DU BERÄKNA";:INPUT N
160 PRINT :PRINT
170 FOR K=1 TO N
180 PRINT:PRINT "VILKET FÖDOÄMNE";:INPUT F$(K)
190 PRINT "HUR MYCKET I GRAM";:INPUT V(K)
200 NEXT K
210 L=L+1
220 READ A$,B
230 IF F$(L)=A$ THEN KAL(L)=B:RESTORE:GOTO 210
240 IF B=-1 THEN PRINT "ÄMNET FINNS EJ":RESTORE:GOTO 210
250 IF L>N THEN 270
260 GOTO 220
270 FOR K=1 TO N
280 TOT(K)=V(K)*KAL(K)/100:SUM=SUM+TOT(K)
290 NEXT K
300 FOR K=1 TO N
310 PRINT:PRINT F$(K);" GER ";INT(TOT(K));" KALORIER"
320 NEXT K
330 PRINT:PRINT "TOTALT BLIR DET ";INT(SUM);" KALORIER"
340 RESTORE:PRINT :PRINT "TRYCK PÅ RETURN";
350 INPUT Q$:GOTO 30
360 CLS
370 READ A$,B:R=R+1
380 IF B=-1 THEN 420
390 PRINT A$,B
400 IF R/15=INT(R/15) THEN PRINT;:INPUT Q$
410 GOTO 370
420 PRINT:PRINT "TRYCK PÅ RETURN";:INPUT Q$
430 CLS:RESTORE
440 CLS:GOTO 140
450 CLS
460 PRINT :PRINT:PRINT "BANTNINGSPROGRAM"
470 PRINT:PRINT "NU KOMMER EN DEL FRÅGOR"
480 PRINT "SOM DU SKALL SVAR PÅ"
490 PRINT:PRINT "HUR MYCKET VÄGER DU";:INPUT KG
500 PRINT "HUR LÅNG ÄR DU ANGE I METER";
510 INPUT L
520 REM ********************UTRÄKNING AV TJOCKVÄRDE
530 TJ = KG / (L) ^ 2
540 IF TJ < 25 THEN D$ = "DU HAR INGET ATT OROA DIG FÖR"
550 IF TJ > 25 AND TJ < 30 THEN D$ = "DU BÖR NOG VÄGA LITE MINDRE"
560 IF TJ > 30 THEN D$ = "SÄTT OMEDELBART IGÅNG MED ATT BANTA"
570 PRINT:PRINT "HUR AKTIV ÄR DU?"
580 PRINT :PRINT "1. STILLASITTANDE"
590 PRINT "2. NÅGOT AKTIV"
600 PRINT "3. MÅTTLIGT AKTIV"
610 PRINT "4. MYCKET AKTIV"
620 PRINT :PRINT"SKRIV SIFFRAN FÖR AKTIVITETSGRAD";:INPUT AK
630 IF AK=1 THEN AK=31
640 IF AK=2 THEN AK=33
650 IF AK=3 THEN AK=35
660 IF AK=4 THEN AK=37
670 KAL=AK*KG
680 CLS
690 PRINT:PRINT "DITT KALORIBEHOV PER DAG"
700 PRINT "FÖR ATT BEHÅLLA DIN VIKT ÄR =";KAL
710 PRINT:PRINT "HUR MYCKET VILL DU GÅ NER PER VECKA"
720 PRINT "UTTRYCKT I KG"; :INPUT NE
730 MIN=NE*1000
740 PRINT :PRINT "FRÅN OCH MED NU FÅR DU BARA"
750 PRINT "ÄTA FÖLJANDE ANTAL KALORIER"
760 PRINT "PER VECKA FÖR ATT GÅ NER";NE;"KG"
770 PRINT :PRINT "VECKA 1.";KAL-MIN;"KALORIER"
780 KG=KG-NE:KAL=AK*KG
790 PRINT "VECKA 2.";KAL-MIN;"KALORIER"
800 KG=KG-NE:KAL=AK*KG
810 PRINT "VECKA 3.";KAL-MIN;"KALORIER"
820 KG=KG-NE:KAL=AK*KG
830 PRINT "VECKA 4.";KAL-MIN;"KALORIER"
840 KG=KG-NE:KAL=AK*KG
850 PRINT "VECKA 5.";KAL-MIN;"KALORIER"
860 KG=KG-NE:KAL=AK*KG
870 PRINT :PRINT:PRINT "OM DU FÖLJER DETTA FÖRSLAG"
880 PRINT "KOMMER DU ATT EFTER 5 VECKOR"
890 PRINT:PRINT "VÄGA";INT(KG*10+.5)/10;"KG"
900 PRINT : PRINT " DITT VÄRDE BLEV ="; INT(TJ*100+.5)/100 ; " "; D$
910 PRINT :PRINT "TRYCK PÅ RETURN";:INPUT Q$
920 GOTO 30
930 CLS:PRINT:PRINT:PRINT TAB(20);"T A C K   F Ö R   I D A G ":END
1000 DATA GLASS,225,MJÖLK,62,SMÖR,748
1010 DATA GRÄDDE,142,OST,400,CORNFLAKES,387
1020 DATA HAVREGRYN,400,VETEMJÖL,350,RIS,341
1030 DATA MAZARINER,479,TUNNBRÖD,375,FRANSKBRÖD,277
1040 DATA KNÄCKEBRÖD,371,MAKARONER,357,WIENERBRÖD,449
1050 DATA MOROT,35,POTATIS,95,RÖBETA,47,AVOCADO,27
1060 DATA BLOMKÅL,25,CHAMPINJON,13,GURKA,12,LÖK,36
1070 DATA LINSER,340,MAJS,83,PAPRIKA,32,PURJOLÖK,26
1080 DATA TOMAT,20,VITKÅL,32,ÄRTER,73,RABARBER,13
1090 DATA SALLAD,17,SOJABÖNOR,403,SPENAT,21
1100 DATA BANAN,93,APELSIN,49,GRAPEFRUIT,35,HALLON,48
1110 DATA JORDGUBBAR,39,KIWI,660,PÄRON,57,ÄPPLE,62
1120 DATA BIFF,200,LEVER,132,KYCKLING,158,FALUKORV,250
1130 DATA FLÄSKFILE,111,SKINKA,250,KOTLETT,203
1140 DATA ENTRECOTE,200,KASSLER,230,WIENERKORV,270
1150 DATA GÄDDA,79,SPÄTTA,88,SILL,219,TORSK,81
1160 DATA ÅL,287,ÄGG,82,APELSINJUICE,50,KAFFE,12
1170 DATA LÄSK,40,SAFT,246,TE,0,ÖL,35,CHOKLAD,578
1180 DATA HONUNG,337,SOCKER,410
1190 DATA -1,-1
1200 DATA -1,-1
```

Så räknar datorn ut kalorimängder

Den här veckan kommer kaloriprogrammet med en sorteringsrutin, så nu hoppas jag att Bengt S i Trelleborg blir nöjd. Jag har inte lagt in någon kosmetik i programmet för att inte göra det för långt.



HEMDATORER

För att snygga upp det så lägg in lite tabuleringar. Du kan skriva PRINT:PRINT TAB (10); i stället för ett enkelt PRINT. Det gör att det ser lite snyggare ut. Har du färg och ljud till din dator så kan du göra det hela lite färggladare. Nu till själva programmet.

Förklaringar

10—20 dimensionering av de indexerade variabler som använts i programmet. S får värdet 100, vilket betyder att jag kan lagra 100 födoämnen. Om du vill ha fler så måste du öka S för att inte få ett felmeddelande.
30—110 menydelen med val till vad du skall göra.
120—160 frågar efter hur många födoämnen du vill beräkna.
170—290 datorn letar reda på dina födoämnen och hur många kalorier de innehåller. Lägg märke till RESTORE i raderna 260 och 270. RESTORE gör att datorn kan läsa hela datalistan med alla fördoämnen från början. När datorn läser i DATA-satserna så bockar den av efter hand så att du inte kan läsa en redan genomgången DATA-sats. För att du skall slippa komma ihåg i vilken ordning födoämnena befinner sig i DATA-satserna så använder man RESTORE för då kan man läsa alla DATA-satserna från början.
300—420 här räknar datorn ihop den sammanlagda kalorimängden för dina födoämne.

Bokstavsordning

430—630 datorn sorterar alla födoämnen i bokstavsordning. Om du vill ha dem sorterade i omvänd ordning så ändra tecknet <till> i rad 520. Variablerna X$ och X är slask-variabler för att lägga värdet från X$(I) och X(I) innan det läggs i variablerna X$(J) och X(J). Värdet måste ligga någonstans i datorn för datorn kan inte hålla värdet i luften. I raderna 570 till 600 skriver datorn ut listan i bokstavsordning.
1000—1190 DATA-satser med namnen på födoämnen och hur många kalorier de innehåller. Om du vill lägga till fler födoämnen så fortsätter du på rad 1190 och tar bort -1,-1. När du är klar så måste du lägga till en rad med -1,-1 för att programmet skall veta var slutet på listan finns.
10000 END slutet på programmet. Om du tittar i menyn på raderna 40 till 100, så ser du att i rad 70 står 3.
AVSLUTA. I rad 100 har du som tredje val rad 10000. Konstruktionen ON... GOTO är bättre att använda i stället för en massa IF... THEN-satser.

Konvertering

Det kommer ibland frågor på en del udda kommando i olika basicdialekter. Jag skall försöka besvara dem, men det kan ibland ta lång tid beroende på platsbrist. Jag fick ett par sådana frågor av Torsten P i Malmö. Det gäller kommandona CDBL, CSNG och CINT i GW-Basic. CDBL konverterar ett enkelt decimaltal till ett med upp till 16 decimaler, så kallad "double-precision". Skrives PRINT CDBL(TAL).
CSNG konverterar ett tal från "double-precision" till ett enkelt tal.
CINT ger heltalsdelen av ett tal och höjer om decimaldelen är över 0.5. Jämför med INT, som inte höjer utan bara plockar ut heltalsdelen.

Skriv följande och testa:
10 PRINT 268.78,CDBL(268.78)
20 PRINT 267.3400345678, CSNG(267.3400345678)
30 PRINT CINT(45.99),INT(45.99)
Då ser du skillnaden på de olika kommandon.
Lycka till med kaloriprogrammet och ha en bra vecka!

```
10 S = 100
20 DIM F$(S), V(S), KAL(S), TOT(S), A$(S), B(S)
30 CLS : PRINT : PRINT
40 PRINT "KALORIEBERÄKNINGSPROGRAM"
50 PRINT "1. LISTA PÅ FÖDOÄMNEN"
60 PRINT "2. BERÄKNA KALORIER"
70 PRINT "3. AVSLUTA"
80 PRINT : PRINT
90 INPUT "VAD ÖNSKAS"; Q
100 ON Q GOTO 430, 120, 10000
110 GOTO 30
120 PRINT
130 PRINT "HUR MÅNGA FÖDOÄMNEN"
140 PRINT "VILL DU BERÄKNA";
150 INPUT N
160 PRINT : PRINT
170 FOR K = 1 TO N
180 PRINT
190 PRINT "VILKET FÖDOÄMNE";
200 INPUT F$(K)
210 PRINT "HUR MYCKET I GRAM";
220 INPUT V(K)
230 NEXT K
240 L = L + 1
250 READ A$, B
260 IF F$(L) = A$ THEN KAL(L) = B: RESTORE: GOTO 240
270 IF B = -1 THEN PRINT "ÄMNET FINNS EJ": RESTORE: GOTO 240
280 IF L > N THEN 300
290 GOTO 250
300 FOR K = 1 TO N
310 TOT(K) = V(K) * KAL(K) / 100
320 SUM = SUM + TOT(K)
330 NEXT K
340 FOR K = 1 TO N
350 PRINT
360 PRINT F$(K); " GER "; INT(TOT(K)); " KALORIER"
370 NEXT K
380 PRINT
390 PRINT "TOTALT BLIR DET "; INT(SUM); " KALORIER"
400 PRINT "TRYCK PÅ RETURN"
410 INPUT QQ$
420 GOTO 30
430 CLS : PRINT "SORTERAR LISTAN ÖVER FÖDOÄMNEN"
440 T = 0
450 T = T + 1
460 READ A$(T), B(T)
470 IF A$(T) = "-1" THEN 490
480 GOTO 450
490 T = T - 1
500 FOR I = 1 TO T - 1
510 FOR J = I + 1 TO T
520 IF A$(I) < A$(J) THEN 540
530 X$ = A$(I): A$(I) = A$(J): A$(J) = X$
540 X = B(I): B(I) = B(J): B(J) = X
550 NEXT J
560 NEXT I
570 FOR I = 1 TO T
580 PRINT TAB(10); A$(I); TAB(25); B(I)
590 IF I / 15 = INT(I / 15) THEN INPUT QQ$: GOTO 600
600 NEXT I
610 PRINT "TRYCK PÅ RETURN"
620 INPUT QQ$
630 RESTORE: GOTO 30
1000 DATA GLASS,225,MJÖLK,62,SMÖR,748
1010 DATA GRÄDDE,142,OST,400,CORNFLAKES,387
1020 DATA HAVREGRYN,400,VETEMJÖL,350,RIS,341
1030 DATA MAZARINER,479,TUNNBRÖD,375,FRANSKBRÖD,277
1040 DATA KNÄCKEBRÖD,371,MAKARONER,357,WIENERBRÖD,449
1050 DATA MOROT,35,POTATIS,95,RÖBETA,47,AVOCADO,27
1060 DATA BLOMKÅL,25,CHAMPINJON,13,GURKA,12,LÖK,36
1070 DATA LINSER,340,MAJS,83,PAPRIKA,32,PURJOLÖK,26
1080 DATA TOMAT,20,VITKÅL,32,ÄRTER,73,RABARBER,13
1090 DATA SALLAD,17,SOJABÖNOR,403,SPENAT,21
1100 DATA BANAN,93,APELSIN,49,GRAPEFRUIT,35,HALLON,48
1110 DATA JORDGUBBAR,39,KIWI,660,PÄRON,57,ÄPPLE,62
1120 DATA BIFF,200,LEVER,132,KYCKLING,158,FALUKORV,250
1130 DATA FLÄSKFILE,111,SKINKA,250,KOTLETT,203
1140 DATA ENTRECOTE,200,KASSLER,230,WIENERKORV,270
1150 DATA GÄDDA,79,SPÄTTA,88,SILL,219,TORSK,81
1160 DATA ÅL,287,ÄGG,82,APELSINJUICE,50,KAFFE,12
1170 DATA LÄSK,40,SAFT,246,TE,0,ÖL,35,CHOKLAD,578
1180 DATA HONUNG,337,SOCKER,410
1190 DATA -1,-1
10000 END
```

Så kan du spela på en enarmad bandit

```
10 CLS
20 PRINT : PRINT "DEN ENARMADE BANDITEN"
30 PRINT : PRINT "DU FÅR 1000 KRONOR NÄR DU STARTAR"
40 PENG = 1000
50 FOR I = 1 TO 10000: NEXT I
60 CLS : PRINT : PRINT "HUR MYCKET SATSAR DU"
70 PRINT : PRINT
80 INPUT "SKRIV ETT TAL OCH TRYCK PÅ RETURN"; A
90 PENG = PENG - A
100 PRINT : PRINT
110 PRINT : PRINT "TRE AV SAMMA SIFFRA GER "; 3 * A
120 PRINT : PRINT "TVÅ AV SAMMA SIFFRA GER "; 2 * A
130 PRINT : PRINT
140 INPUT "TRYCK PÅ RETURN"; Q$
150 FOR K = 1 TO INT(RND * 10 + 3): CLS
160 PRINT : PRINT : PRINT : PRINT
170 Y = INT(RND * 10)
180 Z = INT(RND * 10)
190 Q = INT(RND * 10)
200 PRINT TAB(15); Y; TAB(25); Z; TAB(35); Q
210 FOR I = 1 TO 250: NEXT
220 PRINT : PRINT
230 NEXT
240 INPUT "TRYCK PÅ RETURN"; Q$
250 IF Y = Z THEN 270
260 GOTO 350
270 IF Z = Q THEN 310
280 PRINT : PRINT "DU VANN"; 2 * A
290 PENG = PENG + 2 * A
300 GOTO 440
310 PRINT : PRINT "DU VANN";
320 PRINT 3 * A
330 PENG = PENG + 3 * A
340 GOTO 440
350 IF Y = Q THEN 400
360 IF Z = Q THEN 400
370 PRINT : PRINT "DU VANN INGET"
380 FOR I = 1 TO 1000: NEXT
390 GOTO 440
400 PRINT : PRINT "DU VANN"; 2 * A
410 PENG = PENG + 2 * A
420 PRINT : PRINT
430 PRINT : PRINT
440 PRINT : PRINT "VILL DU FORSÄTTA!"
450 PRINT : PRINT
460 IF PENG < 1000 THEN PRINT "DU HAR HITILLS FÖRLORAT "; 1000 - PENG
470 PRINT : PRINT
480 IF PENG > 1000 THEN PRINT "DU HAR HITILLS VUNNIT"; PENG - 1000
490 PRINT : PRINT
500 INPUT "SKRIV J ELLER N OCH TRYCK PÅ RETURN"; A$
510 IF A$ = "J" OR A$ = "j" THEN 60
520 IF A$ = "N" OR A$ = "n" THEN END
530 GOTO 440
```


Christian Lindeberg

HEMDATORER

Den här veckan skall jag skriva lite om spel. En fråga som dyker upp ofta är: "Hur svårt är det att göra ett spel?." Om man menar ett riktigt professionellt spel så är svaret ungefär två manår.

Det betyder att det skulle ta en man två år att göra ett spel. Nu är det sällan att en man gör jobbet utan det är en hel stab av arbetare, som gör olika saker.

Ett spel gör man på samma sätt, som man gör en film. Först skriver man ner idén sedan gör man ett scenario. I scenariot beskriver man i detalj vad som skall ske i spelet.

Nu börjar det stora arbetet med att planera uppbyggnaden och alla händelser. Mycket ofta använder programmerarna sig av stordatorer för att göra spelen och sedan överföra dem till olika koder för olika datorer. Att göra spel i dag är en industri och inget för den normale programmeraren.

Men du kan göra en hel del enkla spel på din dator beroende på vilka grafik- och ljudmöjligheter du har.

Jag har skrivit ett litet program om den enarmade banditen, som jag skall förklara en del.

Många skriver och vill ha lite utförligare förklaringar till programmen och i dag skall ni få mera kommentarer om programmet.

Experimentera

Programmet är skrivet i GWBasic och jag använder inga speciella kommandon som bara finns i GWBasic. Vissa kommandon kan ha ett annat namn i din basicdialekt.

Rad 10 CLS betyder. CLear Screen, tömmer skärmen och sätter markören längst upp till vänster.

Om du kör med GWBasic, så bör du lägga till en rad 15, som skall se ut så här:
15 RANDOMIZE (TIMER)

Raden behövs för att slumpvalsgeneratorn skall starta på olika tal. Om du inte har rad 15 kommer alltid samma talserie fram efter det du har skrivit RUN.

Raderna 20 och 30 är PRINT-satser, som skriver ut meddelandet mellan citationstecknen.

I rad 40 får variabeln PENG värdet 1000.

Rad 50 är en fördröjningsloop. Datorn står här och snurrar utan att utföra något bara för att du skall få en chans att läsa raderna 20 och 30. Experimentera med att ändra värdet 10000 och se effekten.

I rad 60 startar jag med ett CLS. Är loppen i rad 50 för kort hinner jag inte att läsa i raderna 20 och 30. Efter CLS kommer en PRINT-sats.

Att jag har två PRINT efter varandra i rad 70 gör att jag får två extra radframmatningar.



Speciell loop

Rad 80 har en INPUT-sats för att jag vill ha ett värde i variabeln A. I en del dialekter kan man inte ha någon utskrift i INPUT-satsen i så fall kan du skriva på följande sätt istället:
80 PRINT "SKRIV ETT TAL OCH TRYCK PÅ RETURN";
85 INPUT A

Semikolonet i slutet på rad 80 gör att när jag kör programmet kommer INPUT-frågan att komma direkt efter PRINT-satsen. I rad 90 drages insatsen från de antal poäng eller kronor du hade från början.

I raderna 100 till 140 skriver T ut förklarande text och rad 140 håller programmet på rad 140 tills du har tryckt på RETURN. Raden kan skrivas om som rad 80.

Rad 150 är lite speciell. Det är en loop, som inte har ett bestämt antal gånger den skall snurra utan detta bestäms av delen INT(RND*10+3). RND ger ett tal mellan 0 och 0.999999, detta tal multipliceras med 10 och sedan lägger jag till 3. INT ger heltalsdelen och skulle RND*10 bli noll så måste jag lägga till tre för annars skulle loopen inte snurra. Man kan naturligtvis ha ett fast värde hela tiden men det är inte så roligt.

Rad 160 bara radframmatningar.

Värdena = banditen

170, 180 och 190 ger variablerna Y, Z och Q ett värde mellan 0 och 9. Det är dessa värden som är själva banditen.

Värdena skrivs sedan ut i rad 200 och rad 210 ger dig en chans att se siffrorna.

220 är radframmatningar och rad 230 tar fram en ny loop. Från rad 250 till rad 360 testar datorn om några av värdena är lika och räknar ut hur mycket du har vunnit.

Om du till exempel har fått 7 tre gånger så händer följande. I rad 250 ser datorn att Y och Z är lika och då hoppar den till 270 där den jämför om Z och Q är lika och sedan hoppar den till rad 310 och talar om att du har vunnit tre gånger insatsen.

På samma sätt testar den av de andra kombinationerna.

Har du inga siffror som är lika så hoppar datorn från rad 260 till rad 350 och sedan 360. Därefter följer rad 370 som talar om att du inte har vunnit något.

Skyddsnät

Raderna 400 och 410 är hopp från raderna 350 och 360, som också testar kombinationer av Z, Y och Q.

Resten av programmet frågar om du vill fortsätta och hur mycket du har vunnit eller förlorat.

Lägg märke till raderna 510 och 520 att jag där testar av både litet och stort tecken på J och N.

I rad 530 finns ett skyddsnät, så skulle du trycka på en annan bokstav så kommer du tillbaka till rad 440.

Försök själv att snygga till programmet med de möjligheter din dator har.

Ha en bra vecka!!

Tips för Spader

Den här veckan kommer det mer spel. Jag fick ett brev ifrån Nils A i Landskrona, som frågade efter program till kortspel. Ett program som kunde lägga fyra bridgehänder och ta ut fyra kort till spelet Spader. Jag visste inte vad Spader var för något förrän jag tittade på TV4.



HEMDATORER

Det gäller att slumpmässigt ta ut fyra kort och detta är inte speciellt svårt. Jag gjorde ett program, som även tog ut fyra bridgehänder och en pokerhand. Jag skall också visa på några tekniker man kan använda och peka på en del problem som kan dyka upp.

Programmet består av två moment. Ett moment är att skapa kortleken. Det andra momentet är att ta kort till de olika spelen.

Vi börjar med kortleken.

Raderna 10 till 230 gör kortleken, men vi tar raderna var för sig. Rad 10 och 20 bara upplyser om vad som skall hända. En REM-sats hoppar basicprogrammet över.

Rad 30 dimensionerar de variabler vi behöver. Normalt kan man använda indexerade variabler upp till 10. Till exempel A(0) till A(10) utan att behöva tala om det i en dimensionssats.

Oblandad lek

Den finns basicdialekter där man måste dimensionera alla indexerade variabler.

Variabeln V$(13) skall innehålla de 13 valörerna och KORT$(52) är den färdiga kortleken. POK$(4,13) använder jag för de 4 olika bridgehänderna och till slut variabeln TEST$(52) för att testa så att kortet inte är taget förut.

RANDOMIZE(TIMER) i rad 40 används i bland annat GWBasic för att slumpvalsgeneratorn skall starta på olika tal.

I rad 50 står RESTORE, detta gör att man kan använda DATA-satserna flera gånger. Det kommer en ytterligare förklaring längre fram.

Raderna 60 till 80 läser in de fyra färgerna SPADER, HJÄRTER o s v från 210.

Nästa omgång raderna 90 till 110 läser in de 13 valörerna.

Kortleken skapas sedan slutligen i raderna 120 till 190.

I rad 130 får variabeln N värdet 1. Sedan kommer två styccken dubbelloopar, som i variabeln KORT$(1) till KORT$(52) lägger in korten från SPADER 2 till KLÖVER ESS.

```
10 REM KORTLEK
20 REM DIMENSIONERING
30 DIM V$(13), KORT$(52), POK$(4, 13), TEST$(52)
40 RANDOMIZE (TIMER)
50 RESTORE
60 FOR I = 1 TO 4
70 READ F$(I)
80 NEXT I
90 FOR I = 1 TO 13
100 READ V$(I)
110 NEXT I
120 REM SKAPAR KORTLEK
130 N = 1
140 FOR I = 1 TO 4
150 FOR J = 1 TO 13
160 KORT$(N) = F$(I) + " " + V$(J)
170 N = N + 1
180 NEXT J
190 NEXT I
200 REM DATASATSER
210 DATA SPADER, HJÄRTER, RUTER, KLÖVER
220 DATA 2,3,4,5,6,7,8,9,10
230 DATA KNEKT,DAM,KUNG,ESS
240 CLS
250 PRINT : PRINT TAB(10); "DATORN KAN NU GÖRA FÖLJANDE SAKER"
260 PRINT : PRINT TAB(10); "1. FYRA KORT TILL SPELET SPADER"
270 PRINT : PRINT TAB(10); "2. EN POKERHAND"
280 PRINT : PRINT TAB(10); "3. FYRA BRIDGEHÄNDER"
290 PRINT : PRINT TAB(10); "4 AVSLUTA"
300 PRINT : PRINT TAB(10); "VAD ÖNSKAR DU";
310 INPUT Q$
320 IF VAL(Q$) < 1 OR VAL(Q$) > 4 THEN 240
330 ON VAL(Q$) GOTO 340, 440, 540, 740
340 CLS
350 FOR I = 1 TO 4
360 A = INT(RND * 52) + 1
370 IF TEST$(A) = "TAGET" THEN 360
380 TEST$(A) = "TAGET"
390 PRINT : PRINT TAB(10); KORT$(A)
400 NEXT I
410 PRINT : PRINT TAB(10); "TRYCK PÅ ENTER";
420 INPUT QQ$
430 GOTO 240
440 CLS
450 FOR I = 1 TO 5
460 A = INT(RND * 52) + 1
470 IF TEST$(A) = "TAGET" THEN 360
480 TEST$(A) = "TAGET"
490 PRINT : PRINT TAB(10); KORT$(A)
500 NEXT I
510 PRINT : PRINT TAB(10); "TRYCK PÅ ENTER";
520 INPUT QQ$
530 GOTO 240
540 CLS
550 FOR I = 1 TO 4
560 FOR J = 1 TO 13
570 A = INT(RND * 52) + 1
580 IF KORT$(A) = " " THEN 570
590 POK$(I, J) = KORT$(A)
600 KORT$(A) = " "
610 NEXT J
620 NEXT I
630 FOR I = 1 TO 12
640 FOR J = I + 1 TO 13
650 IF POK$(1, I) < POK$(1, J) THEN 670
660 X$ = POK$(1, I): POK$(1, I) = POK$(1, J): POK$(1, J) = X$
670 NEXT J: NEXT I
680 FOR I = 1 TO 12
690 FOR J = I + 1 TO 13
700 IF POK$(2, I) < POK$(2, J) THEN 720
710 X$ = POK$(2, I): POK$(2, I) = POK$(2, J): POK$(2, J) = X$
720 NEXT J: NEXT I
730 FOR I = 1 TO 12
740 FOR J = I + 1 TO 13
750 IF POK$(3, I) < POK$(3, J) THEN 770
760 X$ = POK$(3, I): POK$(3, I) = POK$(3, J): POK$(3, J) = X$
770 NEXT J: NEXT I
780 FOR I = 1 TO 12
790 FOR J = I + 1 TO 13
800 IF POK$(4, I) < POK$(4, J) THEN 820
810 X$ = POK$(4, I): POK$(4, I) = POK$(4, J): POK$(4, J) = X$
820 NEXT J: NEXT I
830 FOR I = 1 TO 4
840 FOR J = 1 TO 13
850 PRINT POK$(I, J); "  ";
860 NEXT J
870 PRINT : PRINT
880 NEXT I
890 END
```

Nu har vi alltså en oblandad kortlek. Nu skulle man vilja blanda korten, som vi normalt gör men detta är onödigt för att vi kan få fram samma effekt genom att ta ut korten slumpvis.

Andra delen

Över nu till den andra delen av programmet. Den börjar i rad 240 och består först av en meny i raderna 250 till 330 där man kan välja på fyra olika saker.

1. Få ut fyra kort till spelet SPADER.
2. Fem kort till en pokerhand.
3. Fyra bridgehänder.
4. Avsluta.

I rad 310 hamnar den siffra du har valt från menyn, men eftersom jag har använt en strängvariabel, alltså bokstäver, måste jag med hjälp av funktionen VAL() föra över siffran till ett numeriskt värde. Om du har valt siffran 1 så hamnar den som tecknet "1" i datorn och med hjälp av VAL("1") blir det det numeriska talet 1. Det kan vara bra att kunna detta trick för ibland behöver man denna överföring. Det finns också en funktion för att gå åt andra hållet och den heter STR$. Om du har ett tal till exempel 5 och vill överföra det till tecknet 5 så skriver du så här STR$(5), så kommer talet att lagras som tecknet "5" i datorn. Dessa tekniker är bra till exempel när man skall plocka ut entals, tiotals, hundratals o s v siffror ur olika tal.

Ren skärm

Åter till korten. Rad 320 testar så att jag inte tryckt in en felaktig siffra och i 330 hoppar programmet till den rutin, som jag har valt.

Den första rutinen startar i rad 340 med att rengöra skärmen. I den följande loopen på raderna 350 till 400 tar programmet ut 4 stycken kort.

I rad 360 tar slumpvalsgeneratorn ut ett tal mellan 1 och 52. Jag har använt två olika sätt för att kontrollera att jag inte tar samma kort två gånger.

I raderna 370 och 380 ser ni den tekniken. När en siffra är tagen så skapas en variabel TEST$(siffra), som innehåller ordet "TAGET". När nästa kort kommer fram så testar programmet i rad 370 att det inte finns en variabel med motsvarande nummer och ordet "TAGET". Om det finns en sådan variabel går programmet tillbaka till raden 360 för att få en ny siffra mellan 1 och 52.

I rad 390 skrivs korten ut.

Nästa del, som är fem kort till en pokerhand ser likadan ut som föregående.

Tömmer rad 60

Den sista delen av programmet, de fyra bridgehänderna, är liknande de två övriga men här skapar jag en variabel POK$(I,J) där I är en siffra mellan 1 och 4 och J en siffra mellan 1 och 13. Detta kommer att motsvara de olika händerna. POK$(1,1) till POK$(1,13) blir den första handen. Sedan blir POK$(2,1) till POK$(2,13) den andra o s v.

Här har jag använt en annan teknik för att kontrollera om kortet är taget. Alla korten ligger i variablerna KORT$(1) till KORT$(52). Efterhand som korten delas ut tömmer jag innehållet i rad 600. I rad 590 lägger jag in korten i de olika händerna. Det enda som återstår är att sortera korten i de olika händerna, vilket sker i raderna 630 till 820. Utskriften kommer sedan i raderna 830 till 880. Fundera över varför programmet inte sorterar helt rätt. I nästa vecka skall jag fortsätta med fler förklaringar. Ha en bra vecka!

Förra veckans program uppdelat i subrutiner



HEMDATORER

Ja, ja, det var ett fel i förra veckans program. Det känns bra att vara så påpassad, så att ni läser och testar mina program meddetsamma. Felet låg i rad 330. Den rätta rad 330 skall se ut så här:

330 ON VAL(Q$) GOTO 340,440,540,890

Det skulle vara 890 på slutet i stället för 740.

Martin M från Malmö påpekade detta, men undrar också om man inte kunde slå ihop de fyra sorteringsrutinerna till en? Jo, det går bra för om du studerar de fyra rutinerna från förra veckan så ser du att den enda skillnaden är att variabeln POK$(1,l) ändrar till POK$(2,l) o s v.

I stället kan man göra en loop, som ger de siffrorna.

Eftersom jag fick ytterligare en fråga från Mats N i Malmö om hur man delar upp programmet i subrutiner, så har jag skrivit om förra veckans program med uppdelning i subrutiner och en komprimerad sorteringsrutin.

Första delen av programmet har jag kallat INITIERA. Där dimensionerar jag alla variabler och sätter igång RANDOMIZE med klockan TIMER. Här hoppar jag från rad 5 först till rad 10 för dimensioneringen, sedan till subrutinen ortleken och till slut till menyn. Nästa subrutin är den som gör kortleken. Raden 30 måste jag ha för att kunna använda DATA-satserna flera gånger. Om du programmerar i Amiga-basic, QBasic eller liknande så sätter du ett RETURN i slutet på kortleksrutinen. Du kan kalla alla rutinerna med de namn som jag har gett r rutinen.

Nästa rutin är menyn och där du välja vilken subrutin du skall fortsätta med.

Jag tar subrutinerna efter hand. Först Spader som ger de fyra korten till spelet Spader. Lägg märke till att varje subrutin måste sluta med ett RETURN.

Nästa är POKER, som nästan är identisk med Spader.

Till slut kommer BRIDGE med den komprimerade sorteringsrutinen.

Rutinen består av tre så kallade nestade loopar. Detta kan du se i raderna 620 till 670. Den första loopen i rad 620 ger värdena till de fyra olika händerna.

Till slut kommer en rutin på en rad nämligen SLUT. Hit hoppar programmet när du vill sluta.

Förra veckan bad jag dig att fundera på varför sorteringen inte blev perfekt utan att den sorterar de klädda korten fel i ordningen ESS, DAM, KNEKT, KUNG. Anledningen är uppenbar då programmet alltid sorterar i bokstavsvärde.

Kan man klara sig ur detta? En del är perfektionister och vill ha perfekta program.

Problemet går att lösa. De, som har bra minne kommer kanske ihåg de problem, som uppkom med ÅÄÖ i sortering och hur vi löste detta.

Enklast gör man det med en liten subrutin, som jag kommer att visa nästa vecka. I denna subrutin kommer jag att använda funktionerna RIGHT$ och MID$. Ta och titta över stränghanteringsfunktionerna till nästa vecka.

```
5 GOSUB 10: GOSUB 30: GOSUB 220
10 DIM V$(13), KORT$(52), POK$(4, 13), TEST$(52)
20 RANDOMIZE (TIMER): RETURN
```

INITIERA

```
30 RESTORE
40 FOR I = 1 TO 4
50 READ F$(I)
60 NEXT I
70 FOR I = 1 TO 13
80 READ V$(I)
90 NEXT I
100 REM SKAPAR KORTLEK
110 N = 1
120 FOR I = 1 TO 4
130 FOR J = 1 TO 13
140 KORT$(N) = F$(I) + " " + V$(J)
150 N = N + 1
160 NEXT J
170 NEXT I
180 REM DATASATSER
190 DATA SPADER, HJÄRTER, RUTER, KLÖVER
200 DATA 2,3,4,5,6,7,8,9,10
210 DATA KNEKT,DAM,KUNG,ESS
215 RETURN
```

KORTLEK

```
220 CLS
230 PRINT : PRINT TAB(10); "DATORN KAN NU GÖRA FÖLJANDE SAKER"
240 PRINT : PRINT TAB(10): "1. FYRA KORT TILL SPELET SPADER"
250 PRINT : PRINT TAB(10); "2. EN POKERHAND"
260 PRINT : PRINT TAB(10); "3. FYRA BRIDGEHÄNDER"
270 PRINT : PRINT TAB(10); "4 AVSLUTA"
280 PRINT : PRINT TAB(10); "VAD ÖNSKAR DU";
290 INPUT Q$
300 IF VAL(Q$) < 1 OR VAL(Q$) > 4 THEN 240
310 ON VAL(Q$) GOSUB 330, 430, 530, 750
320 GOSUB 30 : GOTO 220
```

MENY

```
330 CLS
340 FOR I = 1 TO 4
350 A = INT(RND * 52) + 1
360 IF TEST$(A) = "TAGET" THEN 360
370 TEST$(A) = "TAGET"
380 PRINT : PRINT TAB(10); KORT$(A)
390 NEXT I
400 PRINT : PRINT TAB(10); "TRYCK PÅ ENTER";
410 INPUT QQ$
420 RETURN
```

SPADER

```
430 CLS
440 FOR I = 1 TO 5
450 A = INT(RND * 52) + 1
460 IF TEST$(A) = "TAGET" THEN 360
470 TEST$(A) = "TAGET"
480 PRINT : PRINT TAB(10); KORT$(A)
490 NEXT I
500 PRINT : PRINT TAB(10); "TRYCK PÅ ENTER";
510 INPUT QQ$
520 RETURN
```

POKER

```
530 CLS
540 FOR I = 1 TO 4
550 FOR J = 1 TO 13
560 A = INT(RND * 52) + 1
570 IF KORT$(A) = " " THEN 570
580 POK$(I, J) = KORT$(A)
590 KORT$(A) = " "
600 NEXT J
610 NEXT I
620 FOR K = 1 TO 4
630 FOR I = 1 TO 12
640 FOR J = I + 1 TO 13
650 IF POK$(K, I) < POK$(K, J) THEN 680
660 X$ = POK$(K, I): POK$(K, I) = POK$(K, J): POK$(K, J) = X$
670 NEXT J: NEXT I: NEXT K
680 FOR I = 1 TO 4
690 FOR J = 1 TO 13
700 PRINT POK$(I, J); " ":
710 NEXT J
720 PRINT : PRINT
730 NEXT I
740 PRINT :PRINT TAB(10); "TRYCK PÅ ENTER":
750 INPUT QQ$
760 RETURN
```

BRIDGE

```
770 END
```

Så kontrollerar du ditt personnummer

Stränghantering fortsätter den här veckan och vi visar hur man kan använda dessa funktioner när man testar om ett personnummer har rätt kontrollsiffra.



HEMDATORER

Ett personnummer består av 11 tecken — tio siffror och ett bindestreck. Den tionde siffran är en kontrollsiffra, som kontrollerar att de nio andra siffrorna är riktiga. Om kontrollsiffran är fel så betyder det att någon av de tio siffrorna är felaktiga eller har blivit fel inslagna.

I dag har vi också "personnummer" på matvaror. Den vanligaste är EAN-koden, EAN=European Article Numbering, en streckkod, som vi hittar på nästan alla livsmedel. EAN-koden består av 13 siffror där den sista också är en kontrollsiffra.

De koder som används på varor avläses av en ljuspenna eller en litens canner. Koderna innehåller bland annat artikelnumret på varan. Avläsaren är kopplad till en dator, som kontrollerar att koden stämde med hjälp av kontrollsiffran. Sedan letar datorn upp vad varan kostar och ger denna information tillbaka till kassan. Priset finns alltså inte i etiketten utan i datorn.

Kolla alltid!

Det är den vanligaste missuppfattningen att priset finns i etiketten. Att priset finns i datorn gör att man alltid måste vara noga med att kontrollera att det blir rätt pris man betalar. Det pris, som man har programmerat in i datorn stämmer inte alltid överens med vad som prismärkningen anger.

Jag har ofta kommit på handlare, som har haft extrapriser på varor och när man kommer till kassan så ger datorn det ordinarie priset. Den enda ursäkten man får är att man inte hunnit, eller glömt, att programmera om datorn — så kolla vad ni betalar.

Nu över till kontrollsiffran. En kontrollsiffra får man fram genom att multiplicera alla tal i udda position med 2 och alla i jämn position med 1. Jag skall visa med ett exempel se figur.

Samma system gäller om det är fler siffror. Nu gäller det att sätta in detta i ett program.

Jag har gjort ett program, som är uppdelat på sex subrutiner.

1. Dimensionering av variabler.
2. Läsa in månader.
3. Inmatning av personnummer.
4. Uträkning av kontrollsiffra.
5. Uträkning av månad.
6. Utskrift av resultat.

Månadernas namn

I dimensioneringen finns bara en variabel nämligen M$(12) för månadernas namn. Så fort en variabel skall vara mer än 10, så måste den dimensioneras.

I subrutinen, som börjar på rad 200 läser programmet in de tolv månadernas namn och lägger det variablerna M$(1) till M$(12).

Nästa subrutin tar hand om inmatning av personnummer och kontrollerar i en IF THEN-sats att det inte finns för många eller för lite tecken. Subrutinen som börjar på rad 10000 kontrollerar om det är för lite eller för mycket tecken.

Subrutinen, som börjar på rad 4000, den där det egentligen händer. Jag skall ta denna rutin mera i detalj.

I rad 4010 tar programmet bort kontrollsiffran och bindestrecket med hjälp av funktionen MID$(). Du får en utskrift på skärmen i rad 4030 så att du kan kontrollera att datorn gjort rätt.

I raderna 4050 till 4120 finns en loop som räknar ut siffersumman i personnumret.

I rad 4060 får variabeln N ett numeriskt värde när funktionen MID$() plockar ut tecken för tecken i strängen NP$. Funktionen VAL ger tecknet ett numeriskt värde, eller kort sagt ser till att man kan räkna med värdet i datorn.

Sista kontrollen

Rad 4070 kontrollera positionen på tecknet, om det är i jämn position skall det multipliceras med 1 annars med 2.

Rad 4100 ser kanske lite mystisk ut, men titta i figuren så ser du att 2*7=14. vi skall inte ha talet 14 utan summan av siffrorna som är 1+4=5. Talet 5 får du också fram om du tar talet 14 och minskar med 9.

Den sista viktiga kontrollen är i rad 4160. för om siffersumman i rad 4110 har blivit ett tiotal, så blir resultatet av rad 4150 talet 10 och detta kan inte vara en kontrollsiffra. Därför måste vi göra om 10 till siffran 0, som då blir kontrollsiffra.

Jag fortsätter med resten nästa gång då jag skall lägga till en rutin som talar om vilket kön personen har.

Ha en bra vecka!!!

```
 701216-3999   personnummer
  701216-399
 *212121 212
 1402226 6918
Summan=1+4+0+2+2+2+6+6+9+1+8=41
Ta bort tiotalet då återstår 1.
Nästa steg blir 10 - 1 = 9, som är kontrollsiffran
```

```
10 REM PROGRAMM FÖR ATT BERÄKNA KONTROLLSIFFRA UR PERSONNUMMER
20 REM AV CHRISTIAN LINDEBERG
30 GOSUB 1000: REM DIMENSIONERING
40 GOSUB 2000: REM LÄSA IN MÅNADER
50 GOSUB 3000: REM INMATNING AV PERSONNUMMER
60 GOSUB 4000: REM UTRÄKNING AV KONTROLLSIFFRA
70 GOSUB 5000: REM UTRÄKNING AV MÅNAD
80 GOSUB 6000: REM UTSKRIFT
100 CLEAR : GOTO 10
1000 REM DIMENSIONERING
1010 DIM M$(12)
1020 RETURN
2000 REM LÄSA IN MÅNADER
2010 FOR I = 1 TO 12
2020 READ M$(I)
2030 NEXT I
2040 DATA JANUARI,FEBRUARI,MARS,APRIL,MAJ,JUNI,JULI
2050 DATA AUGUSTI,SEPTEMBER,OKTOBER,NOVEMBER,DECEMBER
2060 RETURN
3000 REM INMATNING AV PERSONNUMMER
3010 CLS
3020 PRINT : PRINT TAB(10); "SLÅ IN ETT PERSONNUMMER"
3030 PRINT : PRINT TAB(10); "MED ALLA SIFFROR OCH BINDESTRECK"
3040 PRINT : PRINT TAB(10); "I FORMMEN 111111-1111"
3050 PRINT : PRINT TAB(10); : INPUT PN$
3060 IF LEN(PN$) < 11 OR LEN(PN$) > 11 THEN GOSUB 10000: GOTO 3000
3070 RETURN
4000 REM BERÄKNING AV KONTROLLSIFFRA
4010 NP$ = MID$(PN$, 1, 6) + MID$(PN$, 8, 3)
4020 PRINT : PRINT TAB(10); "BORTTAGNING AV BINDESTRECK"
4030 PRINT : PRINT TAB(10); "OCH KONTROLLSIFFRA "; NP$
4040 PRINT : PRINT TAB(10); "TRYCK PÅ ENTER"; : INPUT QQ$
4050 FOR I = 1 TO 9
4060 N = VAL(MID$(NP$, I, 1))
4070 IF I / 2 = INT(I / 2) THEN F = 1: GOTO 4090
4080 F = 2
4090 T = F * N
4100 IF T > 9 THEN T = T - 9
4110 S = S + T
4120 NEXT I
4130 B$ = STR$(S)
4140 K$ = MID$(B$, 3, 1)
4150 KONTROLL = 10 - VAL(K$)
4160 IF KONTROLL = 10 THEN KONTROLL = 0
4170 RETURN
5000 REM UTRÄKNING AV MÅNAD
5010 UT$ = MID$(PN$, 3, 2)
5020 MAN = VAL(UT$)
5030 RETURN
6000 REM UTSKRIFT
6010 CLS
6020 PRINT : PRINT TAB(10); "DET INSLAGNA PERSONNUMRET = "; PN$
6030 PRINT : PRINT TAB(10); "DEN UTRÄKNADE KONTROLLSIFFRAN ="; KONTROLL
6040 IF VAL(MID$(PN$, 11, 1)) = KONTROLL THEN 6060
6050 PRINT : PRINT TAB(10); "KONTROLLSIFFRORNA STÄMMER INTE": GOTO 6070
6060 PRINT : PRINT TAB(10); "KONTROLLSIFFRORNA ÄR LIKA"
6070 PRINT : PRINT TAB(10); "PERSONEN ÄR FÖDD"
6080 PRINT : PRINT TAB(10); "DEN "; MID$(PN$, 5, 2); " "; M$(MAN); " 19"; MID$(PN$, 1, 2)
6090 PRINT : PRINT TAB(10); "TRYCK PÅ ENTER ";
6100 INPUT QQ$
6110 PRINT : PRINT TAB(10); "NY BERÄKNING (J/N)";
6120 INPUT QQ$
6130 IF QQ$ = "J" OR QQ$ = "j" THEN 6150
6140 IF QQ$ = "N" OR QQ$ = "n" THEN END
6150 RETURN
10000 CLS
10010 IF LEN(PN$) < 11 THEN 10100
10020 IF LEN(PN$) > 11 THEN 10200
10100 PRINT : PRINT TAB(10); "DU HAR MATAT IN FÖR LITE TECKEN"
10110 PRINT : PRINT TAB(10); "GLÖM INTE BINDESTRECK MELLAN"
10120 PRINT : PRINT TAB(10); "SJÄTTE OCH SJUNDE SIFFRAN"
10130 PRINT : PRINT TAB(10); "TRYCK PÅ ENTER";
10140 INPUT QQ$
10150 RETURN
10200 PRINT : PRINT TAB(10); "DU HAR SLAGIT IN FÖR MÅNGA TECKEN"
10210 PRINT : PRINT TAB(10); "DU FÅR INTE HA NÅGRA MELLANSLAG"
10220 PRINT : PRINT TAB(10); "TRYCK PÅ ENTER";
10230 INPUT QQ$
10240 RETURN
```

Länsprogram



Christian Lindeberg

HEMDATORER

Den här veckan skall jag fortsätta med personnummer och programmet, som räknar ut i vilket län du är född i. Personnumret består av 11 tecken 1 bindestreck och 10 siffror. De sex första siffrorna är födelsedagen och de fyra sista har flera betydelser. Den sista siffran är kontrollsiffran, som vi har testat i de tidigare programmen. Siffra nummer nio talar om det är man eller kvinna. De tre siffrorna efter bindestrecket talar om var du är född.

Här följer en uppställning för de olika länen:
STOCKHOLMS LÄN 001-139
UPPSALA LÄN 140-159
SÖDERMANLANDS LÄN 160-189
ÖSTERGÖTLANDS LÄN 190-239
JÖNKÖPINGS LÄN 240-269
KRONOBERGS LÄN 270-289
KALMAR LÄN 290-319
GOTLANDS LÄN 320-339
BLEKINGE LÄN 330-349
KRISTIANSTADS LÄN 350-389
MALMÖHUS LÄN 390-459
HALLANDS LÄN 460-479
GÖTEBORGS OCH BOHUS LÄN 480-549
ÄLVSBORGS LÄN 550-589
SKARABORGS LÄN 590-619
VÄRMLANDS LÄN 620-659
ÖREBRO LÄN 660-689
VÄSTMANLANDS LÄN 690-709
KOPPARBERGS LÄN 710-749
GÄVLEBORGS LÄN 750-779
VÄSTERNORRLANDS LÄN 780-819
JÄMTLANDS LÄN 820-849
VÄSTERBOTTENS LÄN 850-889
NORRBOTTENS LÄN 890-929
RIKSSKATTEVERKET 930-999

Om du tittar på de olika ser du siffrorna 930 till 999 tillhör skatteverket och delas ut till länen när deras siffror har tagit slut. Detta sker när det har fötts fler flickor eller pojkar en dag så att alla jämna eller udda kombinationer tagit slut.

Du ser att programmet består av många DATA-satser. I varje DATA-sats finns information om län och om start- och slutsiffra. Jag har delat in programmet i 5 stycken subrutiner. Den första dimensionerar variablerna, den andra läser in alla länen och deras olika värde.

Den tredje subrutinen är en inmatningsrutin för personnumret och den fjärde räknar ut var personen är född. Uträkning av födelselän sker i rad 3050 där siffrorna sju, åtta och nio överförs till ett tal genom funktionen VAL. På samma gång testas i vilket intervall talet ligger, när programmet har hittat rätt intervall så får variabeln B$ namnet på det län där intervallet fanns.

Den sista rutinen är en frågerutin om du skall sluta eller fortsätta. Glöm inte att alltid testa både stora och små bokstäver.

I raderna 3050 och 5030 har jag använt mig av logiska operatorer, nämligen AND och OR.

I rad 3050 testar programmet om de tre siffrorna ligger inom ett visst intervall. Talet måste uppfylla kravet att vara större än ett visst tal och mindre än ett annat tal. Dessa tal kommer ifrån DATA-satserna efter namnet på de olika länen. Lägg märke till att i jämförelsen också finns två jämförelseoperatorer i första delen större än och lika med och efter AND mindre än och lika med.

Rad 5030 testar bara om det är stora N eller lilla n.

I nästa artikel skall jag tala mer om de logiska operatorerna AND, OR och NOT.

Vad händer om man skriver PRINT 2 AND 3 eller PRINT 2 OR 3 eller PRINT 2 NOT 3?

Varför?

Gymnastisera hjärncellerna till nästa vecka.

Ha en bra vecka!

```
10 GOSUB 1000
20 GOSUB 2000
30 GOSUB 3000
40 GOSUB 4000
50 GOSUB 5000
60 CLEAR
70 GOTO 10
1000 CLS
1010 DIM F$(25),START(25),SLUT(25)
1020 RETURN
2000 DATA STOCKHOLMS LÄN,001,139
2010 DATA UPPSALA LÄN,140,159
2020 DATA SÖDERMANLANDS LÄN,160,189
2030 DATA ÖSTERGÖTLANDS LÄN,190,239
2040 DATA JÖNKÖPINGS LÄN,240,269
2050 DATA KRONOBERGS LÄN,270,289
2060 DATA KALMAR LÄN,290,319
2070 DATA GOTLANDS LÄN,320,339
2080 DATA BLEKINGE LÄN,330,349
2090 DATA KRISTIANSTADS LÄN,350,389
2100 DATA MALMÖHUS LÄN,390,459
2110 DATA HALLANDS LÄN,460,479
2120 DATA GÖTEBORGS OCH BOHUS LÄN,480,549
2130 DATA ÄLVSBORGS LÄN,550,589
2140 DATA SKARABORGS LÄN,590,619
2150 DATA VÄRMLANDS LÄN,620,659
2160 DATA ÖREBRO LÄN,660,689
2170 DATA VÄSTMANLANDS LÄN,690,709
2180 DATA KOPPARBERGS LÄN,710,749
2190 DATA GÄVLEBORGS LÄN,750,779
2200 DATA VÄSTERNORRLANDS LÄN,780,819
2210 DATA JÄMTLANDS LÄN,820,849
2220 DATA VÄSTERBOTTENS LÄN,850,889
2230 DATA NORRBOTTENS LÄN,890,929
2240 DATA RIKSSKATTEVERKET,930,999
2250 FOR I=1 TO 25
2260 READ F$(I),START(I),SLUT(I)
2270 NEXT I
2280 RETURN
3000 PRINT:PRINT TAB(10);"SLÅ IN DITT PERSONNUMMER"
3010 PRINT:PRINT TAB(10);"MED BINDESTRECK";
3020 INPUT P$
3030 A$=MID$(P$,8,3)
3040 FOR I=1 TO 25
3050 IF VAL(A$)>=START(I) AND VAL(A$)<=SLUT(I) THEN
     B$=F$(I)
3060 NEXT I
3070 RETURN
4000 PRINT:PRINT TAB(10);"DU ÄR FÖDD I ";
4010 PRINT B$
4020 RETURN
5000 PRINT:PRINT TAB(10);"VILL DU TESTA FLER
     PERSONNUMMER";
5010 PRINT " (J/N)";
5020 INPUT Q$
5030 IF Q$="N" OR Q$="n" THEN END
5040 RETURN
```

Registerprogram

Den här veckan blir det en kort artikel och ett långt program. Anledningen är alla de frågor, som jag fått om registerprogram. Frågorna har varit om register för frimärke, skivor, namn, videofilmer o s v.



HEMDATORER

Jag har gjort ett allmänt program för personuppgifter, som man lätt kan ändra till andra behov. Antalet uppgifter i programmet är sex och fyra av dem är sökbara. En sorteringsrutin sorterar alla personuppgifter efter efternamnet, som ligger i variabeln E$(N). Om du använder registerprogrammet till andra saker och vill göra det enkelt för dig själv så lägg det du vill sortera på i variabeln E$(N).

Det betyder att du måste lägga det först i DATA-satserna.

I rad 10 måste du tilldela variabeln N det antal DATA-satser som börjar i rad 1000. Jag har 12 DATA-satser ger därför N värdet 12. Ordningen på variablerna är E$(N), F$(N),A$(N),P$(N),PA$(N) och T$(N).F$(N) och PA$(N) kan du inte söka på, om du vill söka på F$(N) eller PA$(N) kan du göra rutin, som liknar en av de fyra andra i raderna 190,260,330 och 400.

Programmet startar med att det läser in alla uppgifter i DATA-satserna och placerar dem i sina variabler. Därefter sker en sortering, så att alla uppgifter kommer att ligga sorterade efter innehållet i E$(N). När du lägger till uppgifter så fortsätter du med flera DATA-satser efter rad 1110. Lägger du in många uppgifter, så att raderna innan rad 2000 inte räcker så kan du fortsätta på rad 3000. Men glöm inte att ändra N i rad 10, för annars kommer datorn inte att hitta de nya namnen. Rad 10 styr också dimensioneringen av de olika variablerna i rad 30.

I slutet av programmet har jag lagt en utskriftsrutin för skrivare. Den visar principerna. Om du har en VIC-dator så ta bort REM i rad 2000 och 2060. I raderna 2030 och 2040 tar du bort L i LPRINT, sedan kan du köra programmet. Har du en PC, så kan du köra utskriften direkt. Om du vill ha det snyggare så får du själv leka med utskriften. Du kan använda TAB-funktionen eller SPC-funktionen till utskrifter. Har du en 80-teckens skrivare så kan du skriva ut alla uppgifter på en rad. Jag skall visa med ett exempel. Byt ut 2030 och 2040 mot följande:

```
2030 LPRINT E$I;-"";F$(I);SPC(3);A$(I);SPC(3);P$(-I);"";PA$;SPC(3);T$(I)
```

Vill du har det i snygga kolumner så använd TAB-funktionen i stället.

Sorteringsfunktionen är av typen QUICK-sort. Om du vill sortera på någon annan variabel så måste du ändra i raderna 590, 600 och 620, det är E$() du skall ändra till den variabel du vill sortera på. Raderna 650 till 690 flyttar variablerna så att alla uppgifter hänger tillsammans.

Sökrutinerna och utskriften av uppgifterna på skärmen är upplagda så att om det finns fler personer med samma efternamn, så kommer de att visas efter varandra. Detta gäller alla sökrutinerna. Det är ganska enkelt att lägga till ytterligare en utskriftsrutin till skrivare om du vill kunna skriva ut de uppgifter du söker på.

Om du vill skriva ut på skrivare alla de efternamn du söker så lägg till följande:

```
215 PRINT:PRINT TAB(10);"SKÄRM ELLER PRINTER (S/P)";
216 INPUT SKRIV$
230 IF B$=E$(I) AND SKRIV$="S" THEN GOSUB 830
235 IF B$=E$(I) AND SKRIV$="P" THEN GOSUB 3000
3000 LPRINT E$(1);" ";F$(I);SPC(3);A$(I);SPC(3);P$(I-);" ";PA$(I);SPC(3);T$(I)
3010 RETURN
```

Ha en bra vecka och lycka till!!

```
10 N = 12: REM ANTAL NAMN
20 PRINT : PRINT TAB(10); "          VAR GOD VÄNTA"
30 DIM E$(N), F$(N), A$(N), P$(N), PA$(N), T$(N)
40 GOSUB 920
50 GOSUB 560
60 CLS
70 PRINT : PRINT TAB(10); "   REGISTERMENY"
80 PRINT : PRINT TAB(10); "1. SÖKNING PÅ NAMN"
90 PRINT : PRINT TAB(10); "2. SÖKNING PÅ ADRESS"
100 PRINT : PRINT TAB(10); "3. SÖKNING PÅ POSTNUMMER"
110 PRINT : PRINT TAB(10); "4. SÖKNING PÅ TELEFONNUMMER"
120 PRINT : PRINT TAB(10); "5. UTSKRIFT PÅ SKÄRM"
130 PRINT : PRINT TAB(10); "6. UTSKRIFT PÅ PRINTER"
140 PRINT : PRINT TAB(10); "7. AVSLUTA"
150 PRINT : PRINT TAB(10); : INPUT "VAD VILL DU SÖKA PÅ (1, 2, 3, 4, 5, 6 ELLER 7)"; A
160 IF A < 1 OR A > 7 THEN 60
170 ON A GOTO 190, 260, 330, 400, 470, 2000, 550
180 GOTO 60
190 CLS
200 PRINT : PRINT TAB(10); "VILKET NAMN? SKRIV EFTERNAMN ";
210 INPUT B$
220 FOR I = 1 TO N
230 IF B$ = E$(I) THEN GOSUB 830
240 NEXT I
250 GOTO 60
260 CLS
270 PRINT : PRINT TAB(10); "VILKEN ADRESS SÖKER DU";
280 INPUT B$
290 FOR I = 1 TO N
300 IF B$ = A$(I) THEN GOSUB 830
310 NEXT I
320 GOTO 60
330 CLS
340 PRINT : PRINT TAB(10); "VILKET POSTNUMMER SÖKER DU";
350 INPUT B$
360 FOR I = 1 TO N
370 IF B$ = P$(I) THEN GOSUB 830
380 NEXT I
390 GOTO 60
400 CLS
410 PRINT : PRINT TAB(10); "VILKET TELEFONNUMMER SÖKER DU";
420 INPUT B$
430 FOR I = 1 TO N
440 IF B$ = T$(I) THEN GOSUB 830
450 NEXT I
460 GOTO 60
470 CLS
480 FOR I = 1 TO N
490 PRINT : PRINT TAB(10); E$(I); " "; F$(I),A$(I)
500 PRINT TAB(10); P$(I); "  "; PA$(I), T$(I)
510 IF I / 6 = INT(I / 6) THEN PRINT : PRINT TAB(10); "TRYCK PÅ RETURN": INPUT QQ$
520 NEXT I
530 PRINT : PRINT TAB(10); "TRYCK PÅ RETURN": INPUT QQ$
540 GOTO 60
550 END
560 F = 1: L(1) = 1: R(1) = N
570 L = L(F): R = R(F): F = F - 1
580 I = L: J = R
590 C$ = E$(L)
600 IF E$(I) >= C$ THEN 620
610 I = I + 1: GOTO 600
620 IF E$(J) <= C$ THEN 640
630 J = J - 1: GOTO 620
640 IF I > J THEN 710
650 SLASK$ = E$(I): E$(I) = E$(J): E$(J) = SLASK$
655 SLASK$ = F$(I): F$(I) = F$(J): F$(J) = SLASK$
660 SLASK$ = A$(I): A$(I) = A$(J): A$(J) = SLASK$
670 SLASK$ = P$(I): P$(I) = P$(J): P$(J) = SLASK$
680 SLASK$ = PA$(I): PA$(I) = PA$(J): PA$(J) = SLASK$
690 SLASK$ = T$(I): T$(I) = T$(J): T$(J) = SLASK$
700 I = I + 1: J = J - 1
710 IF I <= J THEN 600
720 IF J - L >= R - I THEN 770
730 IF I >= R THEN 750
740 F = F + 1: L(F) = I: R(F) = R
750 R = J
760 GOTO 800
770 IF L >= J THEN 790
780 F = F + 1: L(F) = L: R(F) = J
790 L = I
800 IF L < R THEN 580
810 IF F > 0 THEN 570
820 RETURN
830 CLS
840 PRINT : PRINT : PRINT : PRINT TAB(10); "   NAMN        : "; E$(I); " "; F$(I)
850 PRINT : PRINT : PRINT TAB(10); "   ADRESS      : "; A$(I)
860 PRINT : PRINT : PRINT TAB(10); "   POSTNUMMER : "; P$(I)
870 PRINT : PRINT : PRINT TAB(10); "   POSTADRESS : "; PA$(I)
880 PRINT : PRINT : PRINT TAB(10); "   TELEFON     : "; T$(I)
890 PRINT : PRINT : PRINT : PRINT TAB(10); "   TRYCK PÅ RETURN";
900 INPUT QQ$
910 RETURN
920 FOR I = 1 TO N
930 READ E$(I), F$(I), A$(I), P$(I), PA$(I), T$(I)
940 NEXT I
950 RETURN
1000 DATA DISKSSON,JUMBO,TURVÄGEN 6,666 66,KRETSBY,11 22 33
1010 DATA DATASSON,FLOPPY,DISKETTVÄGEN 2,999 99, CHIPHÖGEN, 66 99 33
1020 DATA DISKSSON,DUMBO,TURVÄGEN 6,666 66,KRETSBY,11 22 33
1030 DATA DATASSON,NIKLAS,DISKETTVÄGEN 2,999 99, CHIPHÖGEN, 66 99 33
1040 DATA DISKSSON,TURE,TURVÄGEN 6,666 66,KRETSBY,11 22 33
1050 DATA DATASSON,SVEN,DISKETTVÄGEN 2,999 99, CHIPHÖGEN, 66 99 33
1060 DATA DISKSSON,JULLE,TURVÄGEN 6,666 66,KRETSBY,11 22 33
1070 DATA DATASSON,INGRID,DISKETTVÄGEN 2,999 99, CHIPHÖGEN, 66 99 33
1080 DATA DISKSSON,ANNA,TURVÄGEN 6,666 66,KRETSBY,11 22 33
1090 DATA DATASSON,SIV,DISKETTVÄGEN 2,999 99, CHIPHÖGEN, 66 99 33
1100 DATA DISKSSON,ELSA,TURVÄGEN 6,666 66,KRETSBY,11 22 33
1110 DATA DATASSON,ELISABET,DISKETTVÄGEN 2,999 99, CHIPHÖGEN, 66 99 33
2000 CLS
2010 REM OPEN 4,4:CMD4
2020 FOR I = 1 TO N
2030 LPRINT E$(I); " "; F$(I), A$(I),
2040 LPRINT P$(I); "  "; PA$(I), T$(I)
2050 NEXT I
2060 REM CLOSE 4
2070 GOTO 60
```

Lär er använda datorn produktivt

Det här var en svår artikel att skriva. Dels är det den sista och dels låste sig min dator tre gånger, så det här är fjärde gången jag skriver om den. Eftersom jag skriver direkt i datorn utan manus så vet jag inte riktigt vad jag skrev i mitt första försök.



HEMDATORER

Antagligent tycker min dator att det också är lite svårt att inte få leverera dessa artiklar i framtiden. Arbetet har nämligen beslutat att tills vidare lägga avdelningen Hemdatorer på is.

Vid det här tillfället så blickar man gärna tillbaka på alla de artiklar, som jag skrivit. Den första artikeln var den 8 oktober 1983, då jag skulle fylla 13 år till julen. Alltså lite mer än 8 år sedan. Då var den första av IBM's PS-datorer två år gammal, Commodores VIC-20 ungefär lika gammal och man hade lanserat VIC-64, som fortfarande tillhör världens mest sålda datormodell.

Av alla dessa finns bara VIC-64 kvar att köpa, vilket innebär att modellen är unik inom databranschen.

Miljoner färger

PC-datorn i dag liknar inte mycket av de gamla datorerna, som hade en diskettstation på 360 kB och ett internminne på 64kB. I dag har PC-datorerna en hårddisk på mellan 40 och 1000 MB och internminne mellan 2 till 16 MB. Då kunde man använda 4 färger av 8, i dag kan man använda upp till 16,8 miljoner färger.

Tala om utveckling, VIC-64:an står sig fortfarande på hemdatormarknaden. Den första jag köpte 1983 kostade 6.000 kr, i dag kostar samma dator cirka 1.200 kr, vilket är ett bevis på den snabba utvecklingen av elektroniska komponenter. När man 1983 lagade en VIC-64, så kontrollerade man vilken krets som var felaktig. Tog bort komponenten och lödde dit en ny. I dag tar man bort hela kortet och sätter i ett nytt.

Då skrev jag om VIC-20, VIC-64 och ZX-81, alla förträffliga datorer. Den sistnämnda försvann på grund av ekonomiska orsaker hos den engelske tillverkaren Sinclair.

Det första program jag hade i tidningen var ett program för att kontrollera personnummer och jag tänkte avsluta min artikelserie med samma program, men från 1992.

Vilken datakraft!

Men innan jag slutar så vill jag komma med en del funderingar.

Vi befinner oss i ett svårt ekonomiskt läge, som kommer att vara under 1992 och 1993. Vad bör man göra i dåliga tider? Jo, slå vakt om sina resurser för att man skall kunna utnyttja dem när tiderna blir bättre. Jag skall försöka förklara med lite bättre svenska. Om vi tänker efter hur många det finns i det här landet som har datorer, Amiga, Atari, PC-datorer, VIC-64 och alla andra typer.

Tänk efter vilken stor datakraft vi har i våra hem.

Varför inte använda denna datakraft tillsammans med sina hjärnceller för att göra saker, som ni kan använda er av i framtiden för att tjäna pengar till exempel. Lär er att använda er dator produktivt. Det värsta jag vet är dessa spel av typen Nintendo och liknande, som inte ger något annat än förslappning och ingen möjlighet till ett självständigt tänkande.

Tack allihop!

Jag vill sluta med tack till alla de läsare, som jag har haft under dessa drygt 8 år.

Jag skulle kunna räkna upp många, men det är svårt att inte glömma någon. Den mest intressanta utvecklingen är att under de första åren var det uteslutande ungdomar, som skrev till mig. De sista åren har det varit övervägande äldre och pensionärer. Jag har många brev från pensionärer, som har hittat ett nytt mål efter sin pensionering.

Till er alla.

Ha ett fortsatt gott liv och utnyttja er datakraft.

Tack för mig!

PS Ni som fortfarande har frågor om hemdatorer kan skicka dem (med frankerat svarskuvert) till Hemdatorer, Box 6, 236 21 Höllviken. DS

```
10 CLS
20 PRINT "SLÅ IN DITT PERSONNUMMER I FORMEN XXXXXX-XXXX";
30 INPUT P$
40 A$=MID$(P$,1,6)
50 B$=MID$(P$,8,3)
60 C$=A$+B$
70 PRINT "PERSONNUMMER EFTER BORTTAGNING AV MINUSTECKEN ";C$
80 FOR K=1 TO 9
90 D$=MID$(C$,K,1)
100 PRINT "SIFFRORNA EN OCH EN I P-NUMMER ";D$
110 IF K/2=INT(K/2) THEN M=M+VAL(D$):GOTO 150
120 R=2*VAL(D$)
130 IF R>9 THEN R=R-9
140 M=M+R
150 NEXT K
160 PRINT "SUMMAN=";M
170 E$=STR$(M)
180 S=VAL(MID$(E$,3,1))
190 S=10-S
200 IF S=10 THEN S=0
210 T=VAL(MID$(P$,11,1))
220 IF S=T THEN PRINT "PERSONNUMMER OK":END
230 PRINT "PERONNUMMER INTE KORREKT"
```

Hela hans liv är datorer

Arbetets dataexpert Christian Lindeberg har blivit veteran på unga dar - nu firar han tioårsjubileum i spalterna

"Underbarnet" var det en och en annan som kallade Christian Lindeberg, när han den sjätte oktober för precis tio år sedan började skriva dataspalten i Arbetet. Nu har han hunnit bli tjugotvå år, men datorer är fortfarande en mycket stor del av hans liv. Arbetet träffade Christian för att höra hur han har upplevt dessa tio synnerligen händelserika år inom datans värld, och vad han väntar sig av de tio nästa.

AV ÅSA TRULSSON

När Christian Lindeberg började att skriva dataspalten 1983 var han tolv år gammal - eller ska man kanske hellre säga ung. Yngst i Sverige betraktades han nästan som en sensation. Idag har hans rekord fått konkurrens av andra unga "datasnillen", men Christian poängterar att det ändå finns en viss skillnad. I dagens datoriserade samhälle växer ungarna upp med datorer runt omkring sig, medan datorn ännu låg i sin vagga i början av åttiotalet. Trots det blev Christian tidigt bekant med datatekniken. Hans far arbetade med datorer, och båda hans bröder valde att slå sig in på samma bana. På det viset blev datorer en naturlig del av livet hemma hos familjen Lindeberg.

Att Christian skulle arbeta med datorer rådde det väl knappast någon tvekan om. Han funderade egentligen aldrig över vad han skulle göra; det bara blev så. Sedan några år jobbar han i en databutik, och datorerna, ja, dem har han inga planer på att överge.



Christian Lindeberg som ungt datageni ...

ENORM UTVECKLING

Att påstå att datorerna har utvecklats enormt de senaste åren är att ta i i underkant. Det behöver man inte vara dataexpert för att förstå.

-En miniräknare idag klarar av mer än en dator gjorde för tio år sedan, säger Christian.

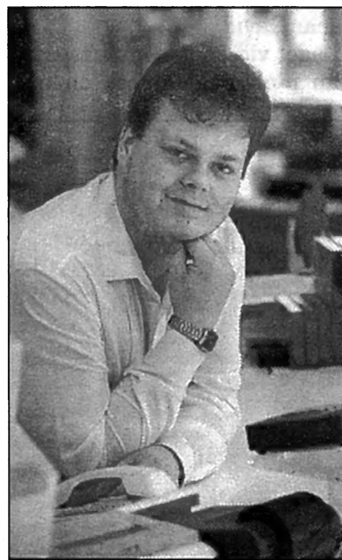
-Det vi idag kallar ordbehandling bestod i princip av att kunna transportera över ord från ett papper till dataskärmen.

Men den största förändringen är kanske ändå datorernas ökade användningsområden. Datorn har förvandlats från något mystiskt och obegripligt till en "var mans pryl". Christian berättar småskrattande om kompisarnas besvikelse när de upptäckte att de stora, blinkande monitors med sladdar kors och tvärs de väntat sig att se hemma hos honom i själva verket mest liknade alldeles vanliga TV-apparater.

Länge möttes både datorer och de som arbetade med dem med stor skepsis och misstänksamhet, något Christian kände på i särskilt hög grad eftersom han dessutom var så ung. Det är först på senare tid,säger han, som det har blivit accepterat, och t.o.m statusbetonat, att syssla med datorer. Han märker också att attityden till dataintresserade ungdomar har förändrats. - När jag började skriva "Dataspalten" kände jag att folk inte riktigt litade på mig, och det är inte förrän de senaste åren jag blivit "vuxen nog" att kunna något. Idag, däremot, är det ingen som ifrågasätter att en femtonåring kanske kan mer om datorer än en vuxen, menar Christian. Skillnader märks också i "Dataspalten" där läsekretsen breddats en hel del. Bl a pensionärer är en grupp på frammarsch.

NÄSTA TIO ÅR?

Det var utvecklingen de senaste tio åren i mycket korta drag - vad med de nästa tio? Kommer datorerna att kunna fortsätta utvecklas i samma hastighet ? - Oh, ja, till och med mer, tror Christian. Något "tak"är definitivt inte nått på länge än.



...och i dag som 20-årig veteran på området.

Om Pc:n var åttiotalets pryl, tror Christian att CD-ROM blir nittiotalets. CD-ROM är ett teknik-CD-program, vilket i korthet innebär att man istället för disketter använder CD-skivor. Fördelarna gentemot disketterna är många; på en CD ryms flera hundra disketter, slitaget blir minimalt samtidigt som det blir en lägre tillverkningskostnad. Precis som CD-skivorna slog ut LP:skivorna, tror han att disketterna successivt kommer att slås ut av "data-CD"-nästa år beräknas hälften av alla nya datorer tilverkas med den här tekniken.

Den största utvecklingsmarknaden framöver tror Christian att hemmet blir. Hans eget experttips för den som i god tid vill skaffa sig det allra senaste är CD 32; en CD-spelare som kopplas till TV:n och styrs av en fjärrkontroll. På den ska man kunna spela sina CD-skivor, eller använda den som uppslagsverk, visa videofilmer på den (givetvis kommer videokasetter också att så småningom helt ersättas av CD) eller kanske flytta över sina foto till foto-CD och göra ett alternativt album. Möjligheterna är oändliga. Universalprylen för varje hem, och så pass billig - 3.500 kr räknar man med - att den har alla möjligheter att bli en konsumentvara.

MÄNNISKAN FRAMFÖR DATORN ...

Christian berättar entusiastiskt om alla nya tekniska framsteg. Själv befinner han sig mitt uppe i den senaste utvecklingen, och hör hela tiden talas om nya tekniska underverk. Händer det aldrig att han känner en viss oro för datorernas oupphörliga utveckling? - Det är inte datorerna och deras möjligheter man ska vara rädd för, utan människorna som sitter framför dem, säger Christian Lindeberg med eftertryck.